Modelli del mondo, architetture a incorporamento congiunto predittivo e sistemi robotici

Uno sguardo ravvicinato su LeWorldModel e il problema della rappresentazione latente stabile

---

Capitolo 1. Intelligenza Autonoma, Percezione Visiva e Razionalità Predittiva

1.1 Il problema epistemologico della previsione

Un agente artificiale che aspiri a operare con qualche forma di competenza in ambienti fisicamente strutturati e temporalmente variabili si trova di fronte a una sfida che precede qualsiasi considerazione tecnica: deve, in un senso che richiede di essere precisato con rigore, *capire* il mondo in cui agisce. Questo non significa, ovviamente, che esso debba disporre di una comprensione filosofica delle proprie circostanze, né che debba elaborare inferenze esplicite sulla natura della realtà. Significa, piuttosto, che deve possedere un modello interno — implicito o esplicito, parametrico o procedurale — che gli consenta di anticipare le conseguenze delle proprie azioni prima ancora di compierle. La distinzione tra sistemi che reagiscono agli stimoli e sistemi che *prevedono* gli esiti è una delle cesure concettuali più significative nell'intera teoria dell'intelligenza artificiale, ed è attorno a essa che si organizza l'intera letteratura dei cosiddetti *modelli del mondo*.

La questione è di natura epistemologica prima che tecnica. Un sistema che opera sulla base di stimoli istantanei — ricevendo un'osservazione, producendo un'azione, ricevendo la successiva osservazione — è fondamentalmente reattivo. Non è in grado di ragionare controfattualmente: non può chiedersi cosa succederebbe se eseguisse una certa azione anziché un'altra, non può simulare mentalmente traiettorie alternative, non può valutare l'utilità a lungo

termine di scelte immediate. La pianificazione, nel senso più ricco del termine, richiede la capacità di simulare internamente porzioni di interazione con il mondo prima che tali interazioni abbiano luogo, selezionando le azioni sulla base delle conseguenze previste piuttosto che su quelle osservate. È questa facoltà — il ragionamento in *spazio immaginativo*, per usare la felice espressione divenuta corrente nella letteratura — che i modelli del mondo mirano a implementare computazionalmente.

Il problema non è nuovo. La teoria del controllo classica, a partire dagli anni Sessanta del Novecento con i lavori di Kalman e Bucy sui filtri ottimali, ha sempre riconosciuto la centralità di un modello interno dell'ambiente nel progetto di controllori efficienti. Il modello di predizione dello stato — che stima, a partire dallo stato corrente e dall'ingresso di controllo, il valore atteso dello stato successivo — è l'ingrediente fondamentale del filtro di Kalman, del controllore predittivo a modello (MPC), e di una vasta classe di algoritmi di controllo ottimale. Ciò che la ricerca sull'intelligenza artificiale moderna ha aggiunto a questa tradizione è essenzialmente una risposta alla domanda: cosa succede quando il modello interno non è costruito analiticamente da un ingegnere, ma deve essere *appreso* da dati osservativi grezzi? E, più specificamente, cosa succede quando questi dati sono sequenze di immagini ad alta dimensionalità, prive di qualsiasi annotazione esplicita sulle variabili di stato rilevanti?

1.2 La visione come fondamento sensoriale privilegiato

La scelta di operare direttamente su immagini — ossia di fondare il sistema sensoriale del modello del mondo sulla percezione visiva — non è arbitraria né puramente pragmatica. È una scelta epistemicamente motivata che ha conseguenze di vasta portata sull'architettura dell'intera catena computazionale.

Le telecamere sono dispositivi economici, scalabili, passivi — non richiedono l'emissione di segnali attivi e non perturbano l'ambiente — e producono un flusso informativo di straordinaria ricchezza circa lo stato dell'ambiente fisico circostante. Un'immagine RGB di dimensione 224×224 pixel contiene $224 \times 224 \times 3 = 150.528$ valori scalari, ciascuno codificante la luminanza di uno dei tre canali cromatici in un punto dello spazio visivo. Questa quantità di informazione è, per la maggior parte dei compiti di controllo robotico, sovra-abbondante: la posizione di un oggetto, la geometria di una scena, la configurazione articolare di un manipolatore sono tutte quantità descrivibili con un numero di parametri di ordine di grandezza inferiore alla dimensione

del pixel-space. Eppure, questa ridondanza è anche una risorsa: la struttura ridondante dei dati visivi consente al sistema di apprendere, attraverso il solo vincolo della coerenza temporale delle sequenze osservate, strutture interne che rispecchiano le proprietà fisiche del mondo.

L'apprendimento direttamente dai pixel — senza l'intermediazione di rappresentazioni di stato pre-costruite manualmente — realizza quello che la comunità chiama apprendimento *end-to-end*: i parametri dell'intero sistema, dall'ingresso sensoriale grezzo all'uscita di controllo, vengono ottimizzati congiuntamente attraverso la retropropagazione del gradiente lungo l'intera catena computazionale. Questa proprietà ha un doppio valore. Da un lato, elimina la necessità di progettare manualmente le feature rilevanti per il compito, lasciando al processo di apprendimento la libertà di scoprire autonomamente le astrazioni più utili. Dall'altro, consente al sistema di adattarsi a domini sensoriali per i quali non esistono modelli analitici affidabili — ambienti deformabili, oggetti trasparenti, superfici riflettenti, condizioni di illuminazione variabile — dove la progettazione manuale delle feature sarebbe intrattabile o impossibile.

Va tuttavia riconosciuto che l'apprendimento da pixel non è privo di costi. La dimensionalità elevatissima dello spazio delle osservazioni rende il problema di stima statisticamente difficile: il numero di parametri dei modelli che operano direttamente sui pixel è tipicamente dell'ordine di decine o centinaia di milioni, e la quantità di dati necessaria per addestrarli efficacemente è corrispondentemente grande. Questa tensione tra ricchezza sensoriale e complessità statistica è una delle ragioni principali per cui la letteratura ha progressivamente spostato l'attenzione verso rappresentazioni latenti compresse: l'idea che sia possibile apprendere una proiezione dell'osservazione in uno spazio a dimensione ridotta tale da preservare le informazioni causalmente rilevanti per la dinamica e il controllo, scartando il resto.

1.3 La struttura formale del problema di apprendimento

Si può dare una formulazione precisa del problema. Sia Ω lo spazio delle osservazioni — per immagini RGB di dimensione $H \times W$, lo si può identificare con $\mathbb{R}^{H\times W\times 3}$, anche se in pratica i valori sono quantizzati in $[0,1]$ o $[0,255]$. Sia $\mathcal{A}$ lo spazio delle azioni, che per compiti di controllo continuo è un sottoinsieme aperto di $\mathbb{R}^A$ per qualche intero $A \geq 1$. L'ambiente è un processo stocastico sulla coppia (osservazione, azione) che si evolve secondo una distribuzione di transizione sconosciuta

$p(o_{t+1} \mid o_{1:t}, a_{1:t})$

che in generale dipende dall'intera storia passata, ma che sotto l'assunzione markoviana si riduce a

$p(o_{t+1} \mid o_t, a_t)$.

Il dataset offline $\mathscr{D}$ è una collezione di N traiettorie della forma

$\mathscr{D} = \{ \tau^{(i)} \}_{i=1}^{N}$, $\tau^{(i)} = (o_1^{(i)}, a_1^{(i)}, o_2^{(i)}, a_2^{(i)}, \ldots, o_T^{(i)}, a_T^{(i)})$

dove $o_t^{(i)} \in \Omega$ è l'osservazione al passo temporale t della traiettoria i-esima, $a_t^{(i)} \in \mathscr{A}$ è l'azione corrispondente, e T è la lunghezza dell'episodio. Le traiettorie sono raccolte da una politica comportamentale qualsiasi — che si indicherà con β — senza alcun requisito di ottimalità rispetto a un compito specifico.

L'obiettivo dell'apprendimento, in questo contesto, non è la massimizzazione di una ricompensa scalare: non ne è prevista alcuna nel dataset. L'obiettivo è la stima di una funzione di transizione approssimata

$f_\Theta : \Omega \times \mathscr{A} \to \Omega$ (modelli generativi)

oppure, nel paradigma delle rappresentazioni latenti,

$enc_\theta : \Omega \to \mathbb{Z} \subseteq \mathbb{R}^d$ $pred_\varphi : \mathbb{Z} \times \mathscr{A} \to \mathbb{Z}$

tale che la coppia $(enc_\theta, pred_\varphi)$ catturi la struttura delle transizioni ambientali in uno spazio latente compresso $\mathbb{Z}$ di dimensione $d \ll H \times W \times 3$.

La qualità di questa stima non può essere misurata direttamente rispetto alla distribuzione di transizione sconosciuta $p(o_{t+1} \mid o_t, a_t)$, ma solo attraverso proxy calcolabili sui dati: distanze nello spazio delle osservazioni o degli incorporamenti, qualità delle previsioni sulla distribuzione del dataset, trasferibilità delle rappresentazioni a compiti di controllo a valle.

1.4 Tensioni fondamentali: astrazione, collasso e utilità del controllo

La tensione fondamentale che attraversa tutta la letteratura sui modelli del mondo riguarda il grado di astrazione della rappresentazione interna: quanto

dell'osservazione grezza è necessario conservare, e quanto può essere scartato senza perdita di potere predittivo e di utilità per il controllo?

Ai due estremi si collocano, rispettivamente, i modelli generativi e i modelli puramente predittivi in spazi latenti non vincolati. I modelli generativi tendono a conservare tutto, pagando un costo computazionale elevato e rischiando di disperdere la capacità del modello su dettagli irrilevanti per il controllo — texture superficiali, variazioni di illuminazione, artefatti di sfondo. I modelli latenti scelgono consapevolmente di operare in spazi compressi, acquistando efficienza al prezzo di una sfida tecnica non banale: garantire che la rappresentazione compressa non degeneri in qualcosa di computazionalmente inutile. È questa sfida — il problema del *collasso della rappresentazione* — a costituire il nucleo concettuale della difficoltà tecnica che questa monografia esamina.

Il collasso non è un rischio teorico astratto. È una conseguenza strutturale del solo obiettivo di previsione in spazi latenti: come si dimostrerà formalmente nel capitolo dedicato, la minimizzazione della perdita di predizione quadratica $\|\hat{z}_{t+1} - z_{t+1}\|^2_2$ è trivialmente soddisfatta dall'encoder costante $enc_\theta(o) = c$ per ogni o, in corrispondenza del quale la perdita è identicamente nulla. Qualsiasi algoritmo di discesa del gradiente tenderà a convergere verso questa soluzione in assenza di vincoli aggiuntivi, rendendo la regolarizzazione anti-collasso non un accessorio ma un componente strutturale indispensabile di qualsiasi architettura JEPA praticabile.

Il paragrafo che segue illustra come questa tensione si manifesta, nella storia della ricerca, attraverso una successione di approcci che riflettono ciascuno una diversa filosofia di compromesso tra capacità espressiva, efficienza computazionale, stabilità dell'addestramento e generalità applicativa.

1.5 La storia come sequenza di compromessi

La ricerca sui modelli del mondo può essere letta come una successione di scoperte di compromessi non ovvi. Il filone dei modelli generativi — da World Models di Ha e Schmidhuber ai recenti DreamerV4, DIAMOND e OASIS — ha dimostrato che è possibile apprendere modelli predittivi di alta qualità nell'ambito dell'apprendimento per rinforzo, ma al prezzo di richiedere segnali di ricompensa e di una complessità computazionale che cresce con la risoluzione delle immagini. Il filone JEPA — da I-JEPA e V-JEPA per la rappresentazione di immagini e video, a PLDM e DINO-WM per il controllo

robotico — ha dimostrato che l'obiettivo predittivo in spazio latente è sufficiente per produrre rappresentazioni utili per la pianificazione, ma ha dovuto confrontarsi con il problema del collasso in modi via via più sofisticati.

L'articolo che questa monografia esamina — il lavoro su LeWorldModel di Maes, Le Lidec, Scieur, LeCun e Balestriero — costituisce un punto di sintesi particolarmente elegante in questa evoluzione. La sua tesi di fondo è che sia possibile addestrare in modo stabile un'architettura JEPA completa, dall'encoder al predittore, end-to-end a partire dai pixel grezzi, con un solo iperparametro effettivo da ottimizzare e con garanzie formali di non-collasso — un risultato che nessuno dei lavori precedenti aveva conseguito simultaneamente su tutte queste dimensioni.

Capitolo 2. Modelli del Mondo: Fondamenti Formali e Tassonomia

2.1 Il dataset offline e la struttura delle traiettorie

Il punto di partenza formale di qualsiasi approccio basato su modelli del mondo è la definizione precisa del dato osservazionale disponibile. Si è già introdotto il dataset offline $\mathcal{D}$ come raccolta di traiettorie. Vale tuttavia la pena approfondire la struttura di queste traiettorie e le ipotesi implicite che si fanno su di esse.

Sia $\mathcal{O}$ lo spazio delle osservazioni — nel caso visivo, lo spazio delle immagini a colori di dimensione $H \times W \times C$, tipicamente con $H = W = 224$ e $C = 3$, cosicché ogni osservazione è un tensore in $\mathbb{R}^{224\times224\times3}$ — e sia $\mathcal{A}$ lo spazio delle azioni, che nel contesto robotico con controllo continuo si assume essere un sottoinsieme aperto di $\mathbb{R}^A$ per qualche intero $A \geq 1$. In LeWorldModel, le azioni sono vettori continui che codificano i comandi di velocità o posizione dei giunti robotici, senza alcuna struttura discreta.

Un dataset offline $\mathcal{D}$ è una collezione di N traiettorie della forma

$$\mathcal{D} = \{ \tau^{(i)} \}_{i=1}^{N}, \quad \tau^{(i)} = (o_1^{(i)}, a_1^{(i)}, o_2^{(i)}, a_2^{(i)}, \ldots, o_T^{(i)}, a_T^{(i)})$$

dove $o_t^{(i)} \in \mathcal{O}$ denota l'osservazione al passo temporale t della traiettoria i-esima, $a_t^{(i)} \in \mathcal{A}$ è l'azione corrispondente e T è la lunghezza dell'episodio. Il dataset viene raccolto da una politica comportamentale arbitraria — che può essere pseudo-esperta, esplorativa o stocastica — con il solo requisito che le traiettorie coprano sufficientemente la varietà delle dinamiche ambientali di interesse.

In assenza di annotazioni di ricompensa, l'obiettivo non è massimizzare alcun segnale scalare di performance, bensì apprendere una rappresentazione delle transizioni ambientali abbastanza generale e strutturata da risultare adattabile a compiti eterogenei in una fase applicativa successiva. Questa configurazione — denominata *offline e reward-free* — è più ambiziosa della semplice imitazione comportamentale: il sistema non sta cercando di replicare la politica comportamentale β, ma di estrarre dalla traiettorie osservate una comprensione della *struttura causale* dell'ambiente.

Nella pratica sperimentale adottata in LeWorldModel, le traiettorie vengono suddivise in sotto-sequenze di lunghezza fissa $N_{sub} = 4$ frame, estratte come segmenti contigui. Tra frame consecutivi viene applicato un *frame skip* di valore $k = 5$: anziché considerare ogni singola immagine, si raggruppano $k = 5$ azioni consecutive tra due frame successivi in un unico blocco d'azione. Questa operazione riduce la frequenza temporale del modello di un fattore 5, consentendo previsioni a orizzonte più lungo con un numero inferiore di passi autoregressivi — un sotto-orizzonte di $H = 5$ passi del modello corrisponde a 25 passi temporali effettivi dell'ambiente — e riducendo la variabilità delle transizioni osservate, che su scale temporali brevi possono essere dominate da artefatti di movimento rapido poco informativi.

La dimensione del mini-batch durante l'addestramento è $B = 128$. Ogni osservazione è un'immagine 224×224×3, i.e. un tensore di 150.528 valori in virgola mobile. La quantità totale di dati impiegata varia per ambiente: 10.000 episodi di 200 passi per OGBench-Cube e Reacher, 20.000 episodi di 196 passi per Push-T, 10.000 episodi di 92 passi per Two-Room. Ogni modello è addestrato per 10 epoche. Questi numeri, pur non essendo eccezionalmente grandi secondo gli standard attuali, sono sufficienti a produrre rappresentazioni di qualità notevole, grazie all'efficienza intrinseca dell'obiettivo JEPA che non disperde capacità su dettagli irrilevanti.

2.2 Modelli del mondo generativi: formalismo e limiti strutturali

I modelli del mondo generativi apprendono una funzione di transizione che opera direttamente nello spazio delle osservazioni, stimando la distribuzione condizionale

$$p(o_{t+1} \mid o_{1:t}, a_{1:t})$$

attraverso un modello parametrico p_θ. Questa distribuzione viene tipicamente fattorizzata per mezzo di un'architettura a due stadi: un primo modulo — l'encoder — mappa le osservazioni in uno spazio di rappresentazione intermedio, discreto o continuo; un secondo modulo — il decoder — genera la distribuzione sulle osservazioni future a partire da tale rappresentazione. Il modello è addestrato massimizzando la log-verosimiglianza delle osservazioni del dataset, o una sua variante come il *variational lower bound* nel caso di modelli a variabile latente.

Nella versione discreta — adottata ad esempio da IRIS e Δ-IRIS — le osservazioni vengono tokenizzate tramite un codice vettoriale discreto (Vector-Quantized VAE, VQ-VAE), producendo sequenze di token discreti. La distribuzione sulla sequenza futura di token viene poi modellata autoregressivamente da un transformer, esattamente come i moderni modelli linguistici modellano la distribuzione sul token successivo dato il contesto. Il decoder ricostruisce l'osservazione a partire dalla sequenza di token predetti. L'obiettivo di addestramento è la massimizzazione di

$$\log p_\theta(o_{t+1} \mid o_t, a_t) = \log \Sigma_z \, p(o_{t+1} \mid z) \, p_\theta(z \mid o_t, a_t)$$

dove la somma è sui possibili codici discreti z. Nella versione continua — adottata ad esempio da DIAMOND, che sostituisce il decoder deterministico con un modello di diffusione — la distribuzione sulle osservazioni future è una gaussiana o una distribuzione implicitamente definita da un processo di diffusione, e l'obiettivo è la massimizzazione del ELBO:

$$\log p_\theta(o_{t+1} \mid o_t, a_t) \geq \mathbb{E}_{q_\varphi(z \mid o_{t+1})}[\log p_\theta(o_{t+1} \mid z)] - \mathrm{KL}(q_\varphi(z \mid o_{t+1}) \,\|\, p_\theta(z \mid o_t, a_t))$$

Il secondo termine è la divergenza di Kullback-Leibler tra la distribuzione posteriore approssimata $q_\varphi(z \mid o_{t+1})$ e la priori condizionale $p_\theta(z \mid o_t, a_t)$, e funge da regolarizzatore che impedisce al modello latente di codificare informazioni irrilevanti per la predizione futura.

Il limite strutturale di questi modelli risiede nella dimensione dello spazio su cui operano. La decodifica di un'immagine 224×224×3 richiede che il decoder stimi 150.528 valori scalari per ogni passo temporale, il che vincola la qualità della predizione alla capacità del decoder di ricostruire i dettagli percettivi. La pianificazione in questo spazio — che implicherebbe ottimizzare una sequenza di azioni $a_{1:H} \in \mathcal{A}^H$ minimizzando un costo definito sulle osservazioni future $o_{1:H}$ — è computazionalmente improponibile: la valutazione di ciascuna traiettoria candidata richiederebbe la generazione di H immagini complete, ciascuna delle quali costa diversi ordini di grandezza in più rispetto alla generazione di un vettore latente.

La planificazione tramite modelli generativi è perciò mediata: il modello generativo viene usato come simulatore per addestrare offline una politica — come in Dreamer — o per generare rollout immaginativi utilizzati per l'aggiornamento dell'agente — come in IRIS. In nessuno di questi casi il modello generativo viene interrogato online durante l'esecuzione in modo da ottimizzare direttamente una sequenza di azioni, il che limita la flessibilità rispetto alla specificazione di nuovi obiettivi.

2.3 Modelli del mondo latenti: astrazione e compressione

L'approccio alternativo — concettualmente più parsimonioso e computazionalmente più trattabile — consiste nel rinunciare alla ricostruzione esplicita delle osservazioni e apprendere invece una funzione di transizione in uno spazio latente di dimensione $d \ll d_{obs} = H \times W \times C$. Si introduce a tal fine un encoder

$$enc_\theta : \mathcal{O} \to \mathbb{Z}, \mathbb{Z} \subseteq \mathbb{R}^d$$

che mappa ogni osservazione o_t nell'embedding latente $z_t = enc_\theta(o_t) \in \mathbb{R}^d$. Su questo spazio compresso si definisce poi un predittore

$$pred_\varphi : \mathbb{Z} \times \mathcal{A} \to \mathbb{Z}$$

il cui compito è stimare l'embedding del frame successivo: dato z_t e l'azione a_t, il predittore produce $\hat{z}_{t+1} = pred_\varphi(z_t, a_t)$, interpretabile come previsione dell'embedding z_{t+1} che l'encoder assocerà al frame o_{t+1} qualora venisse osservato.

La qualità del modello è misurata dalla distanza tra la previsione $\hat{z}_{t+1}$ e l'embedding reale z_{t+1} nello spazio latente, anziché nello spazio delle

osservazioni. Questo spostamento di prospettiva è profondo: l'obiettivo non è più la fedeltà pixel-wise della previsione, ma la fedeltà della struttura latente. Due osservazioni o_{t+1} e o'_{t+1} che differiscono per dettagli visivamente rilevanti ma causalmente irrilevanti per la dinamica futura — variazioni di illuminazione, texture di sfondo — possono avere incorporamenti latenti molto simili, e questa similarità non costituisce un errore del modello ma una sua proprietà desiderabile.

Con d = 192, lo spazio latente è circa 784 volte più piccolo di un'immagine 224×224×3 espressa in floating point a 32 bit, rendendo i rollout autoregressivi del predittore estremamente rapidi. La pianificazione tramite ottimizzazione della sequenza d'azione viene effettuata interamente in $\mathbb{Z}$, con una funzione di costo definita sugli embedding latenti piuttosto che sui pixel. Come verrà discusso nel capitolo dedicato alla pianificazione, questo consente di completare un intero ciclo di ottimizzazione in meno di un secondo anche su hardware di fascia consumer, avvicinando sensibilmente il regime di controllo in tempo reale.

2.4 Tassonomia formale delle strategie anti-collasso

Prima di descrivere in dettaglio l'architettura JEPA e le sue varianti, è utile costruire una tassonomia formale delle strategie adottate per prevenire il collasso della rappresentazione. Questa tassonomia non è meramente classificatoria: rivela la struttura logica del problema e anticipa la posizione che LeWorldModel occuperà rispetto a ciascuna classe.

Si possono distinguere quattro categorie principali, ordinate per crescente sofisticazione formale.

La prima categoria comprende le strategie *architetturali asimmetriche*, che rompono la simmetria del collasso introducendo asimmetrie strutturali tra il ramo di calcolo del target e quello della previsione. L'EMA e lo stop-gradient appartengono a questa categoria: essi impediscono il collasso non attraverso una penalità esplicita, ma attraverso la dinamica differenziata dei due encoder (online e target), che crea una forma di tensione strutturale contro la soluzione costante. Come si argomenterà nel prossimo capitolo, questa categoria manca di fondamenti teorici solidi e la sua efficacia è empiricamente contingente.

La seconda categoria comprende le strategie basate su *regolarizzazione della covarianza*, tra cui VICReg e le sue estensioni temporali in PLDM. Queste

penalizzano esplicitamente il collasso richiedendo che la varianza empirica di ciascuna dimensione latente superi una soglia e che la covarianza tra dimensioni diverse sia piccola. Formalmente, esse impongono vincoli sulla matrice di secondo momento degli embedding: Cov(Z) ≈ I, dove I è la matrice identità. Rispetto alla categoria precedente, offrono una motivazione più esplicita, ma non garantiscono la convergenza a una distribuzione target ben specificata e richiedono la calibrazione di molteplici iperparametri.

La terza categoria comprende le strategie basate su *encoder pre-addestrati congelati*, come DINO-WM. In questo caso il collasso è prevenuto per costruzione, poiché i parametri dell'encoder non vengono aggiornati durante l'addestramento del modello del mondo. La distribuzione degli embedding è quella determinata dal pre-training, che tipicamente presenta buona diversità e struttura. Il prezzo è la perdita dell'adattabilità dell'encoder al dominio specifico.

La quarta categoria — quella a cui appartiene LeWorldModel — comprende le strategie basate su *matching esplicito di distribuzione*, che impongono che la distribuzione degli embedding converga a una distribuzione target specificata. SIGReg, che forza la convergenza a una gaussiana isotropica tramite il test di Epps-Pulley proiettato su direzioni casuali, appartiene a questa categoria. Rispetto alle precedenti, offre la più forte giustificazione formale: la garanzia di non-collasso è un teorema (per il teorema di Cramér-Wold, riveduto nel prossimo capitolo), non un'osservazione empirica.

2.5 Il Model Predictive Control come paradigma di pianificazione robotica

Il *Model Predictive Control* (MPC) fornisce la cornice formale entro cui i modelli del mondo latenti vengono impiegati per il controllo di sistemi robotici durante la fase di inferenza. Il principio fondante dell'MPC è quello di risolvere, ad ogni istante temporale, un problema di controllo ottimo a orizzonte finito.

Sia z_1 = enc_θ(o_1) l'embedding dell'osservazione corrente e sia z_g = enc_θ(o_g) l'embedding dell'osservazione obiettivo. Il problema MPC è:

$$a^*_{1:H} = \operatorname{argmin}_{a_{1:H} \in \mathcal{A}^H} \mathcal{C}(\hat{z}_H)$$

soggetto a:

$$\hat{z}_{s+1} = \text{pred}_\varphi(\hat{z}_s, a_s),\ s = 1, \ldots, H-1 \quad \hat{z}_1 = z_1$$

dove $\mathscr{C} : \mathbb{Z} \to \mathbb{R}$ è una funzione di costo che misura la distanza tra lo stato latente predetto alla fine del rollout e l'embedding dell'obiettivo. La scelta canonica di $\mathscr{C}$ in LeWorldModel è la distanza quadratica euclidea:

$$\mathscr{C}(\hat{z}_H) = \|\hat{z}_H - z_g\|^2_2$$

Questa scelta è motivata dalla struttura della distribuzione degli embedding indotta da SIGReg: poiché la distribuzione è approssimata a una gaussiana isotropica con varianza unitaria in tutte le direzioni, lo spazio latente ha una geometria approssimativamente euclidea uniforme, e la distanza quadratica è una misura di prossimità ragionevole. In uno spazio con struttura anisotropica o non euclidea, sarebbe più appropriato ricorrere alla distanza di Mahalanobis:

$$\mathscr{C}_M(\hat{z}_H) = (\hat{z}_H - z_g)^\top \Sigma^{-1} (\hat{z}_H - z_g)$$

dove Σ è la matrice di covarianza empirica degli embedding — una scelta che tuttavia richiederebbe la stima e l'inversione di una matrice $d \times d$, costosa per d grande.

La caratteristica fondamentale dell'MPC nella sua versione *receding horizon* è che, dopo aver ottimizzato il piano $a_{1:H}$ e averlo eseguito nell'ambiente reale per K passi (con $K \leq H$), il sistema riprende l'osservazione aggiornata e ripianifica ex novo. Solo le prime K azioni del piano ottimizzato vengono effettivamente eseguite. Questo meccanismo — orizzonte scorrevole — compensa l'inevitabile accumulo di errori di predizione del modello appreso: il predittore è un'approssimazione della reale funzione di transizione dell'ambiente, e le sue previsioni su orizzonti lunghi tendono a divergere dalla traiettoria reale. Rieseguire la pianificazione a partire dallo stato corrente reale permette di correggere tali derive prima che diventino incontrollabili.

Il disaccoppiamento tra apprendimento del modello e pianificazione è una caratteristica saliente di questo approccio e merita di essere sottolineato con forza: i parametri θ dell'encoder e φ del predittore rimangono fissi durante tutta la fase di pianificazione. L'ottimizzazione riguarda esclusivamente la sequenza di azioni $a_{1:H}$. Questa separazione semplifica l'analisi teorica del sistema — il modello del mondo è trattato come una funzione deterministica nota — e garantisce che la qualità del piano sia limitata solo dalla qualità del modello appreso, indipendentemente da eventuali instabilità numeriche nell'ottimizzazione delle politiche.

2.6 La famiglia dei modelli generativi orientati al controllo

Per completare il panorama tassonomico, vale la pena collocare i metodi discussi rispetto alla tradizione dei modelli generativi orientati esplicitamente al controllo. DreamerV3, nella sua formulazione, apprende un modello del mondo ricorrente (RSSM, *Recurrent State Space Model*) che mantiene uno stato ricorrente h_t aggiornato dalla transizione

h_t = f(h_{t-1}, z_t, a_{t-1})

dove z_t è un embedding discreto o continuo dell'osservazione corrente, e h_t cattura l'informazione storica pertinente. Il modello di mondo predice la distribuzione sulla prossima coppia (z_{t+1}, r_{t+1}, d_{t+1}) dove r_{t+1} è la ricompensa e d_{t+1} è il segnale di terminazione. La politica e la funzione di valore vengono ottimizzate attraverso rollout immaginativi in questo spazio ricorrente, attraverso la retropropagazione del gradiente attraverso H passi di simulazione interna.

Il confronto con LeWorldModel è illuminante: DreamerV3 richiede ricompense durante l'addestramento del modello del mondo, opera in un regime online (l'agente interagisce continuamente con l'ambiente), e addestra congiuntamente modello, politica e funzione di valore. LeWorldModel separa nettamente le tre fasi — apprendimento del modello (offline, reward-free), pianificazione (online, goal-conditioned) — a prezzo di non poter sfruttare feedback di ricompensa per specializzare le rappresentazioni. Il vantaggio è la massima flessibilità rispetto ai compiti: lo stesso modello del mondo può essere impiegato per qualsiasi obiettivo goal-conditioned senza riaddestrare nulla.

TD-MPC2 rappresenta un punto di sintesi intermedio: apprende un modello latente orientato al compito in modalità online, con accesso alla ricompensa, ma usa l'MPC con CEM durante l'esecuzione esattamente come LeWorldModel. Il confronto sperimentale riportato nell'articolo originale mostra che TD-MPC2, pur essendo specifico per compito, non supera LeWorldModel in modo sistematico sugli ambienti considerati — un risultato che testimonia dell'efficacia del paradigma reward-free quando le strutture fisiche rilevanti sono catturate adeguatamente dall'obiettivo predittivo.

2.7 Considerazioni sulla natura del paradigma offline e reward-free

Vale infine la pena riflettere più in profondità sulla scelta di operare in un regime completamente offline e reward-free, che non è solo un dettaglio tecnico ma una posizione epistemica precisa.

L'apprendimento offline richiede che il dataset $\mathcal{D}$ sia raccolto prima dell'inizio dell'addestramento e non venga arricchito durante l'apprendimento. Questa condizione è restrittiva: un agente online può eseguire esperimenti mirati per colmare le lacune del proprio modello, mentre un agente offline è vincolato alla distribuzione delle traiettorie nel dataset. Se questo dataset non copre adeguatamente alcune regioni dello spazio degli stati — ad esempio le configurazioni di confine di un ambiente, o le transizioni rare ma critiche — il modello appreso potrebbe avere previsioni di qualità insufficiente in quelle regioni, compromettendo la qualità della pianificazione.

L'assenza di ricompense è invece una scelta più radicalmente motivata. Un modello del mondo reward-free è, per costruzione, agnostico rispetto a qualsiasi obiettivo specifico: le strutture latenti che esso apprende sono determinate esclusivamente dalla coerenza temporale delle osservazioni, non dalla loro utilità per raggiungere determinati stati. Questo garantisce una forma di generalità: lo stesso modello può essere riutilizzato per scopi diversi semplicemente modificando la funzione di costo latente. Ma comporta anche un rischio: il modello potrebbe non catturare le strutture fisiche rilevanti per il controllo se esse non sono salienti dal punto di vista della previsione temporale. L'evidenza sperimentale presentata in LeWorldModel — le analisi di probing delle quantità fisiche e le visualizzazioni dello spazio latente — suggerisce che, per gli ambienti considerati, questo rischio è limitato: l'obiettivo predittivo è sufficiente a indurre rappresentazioni che codificano posizioni, angoli e configurazioni in modo linearmente accessibile. Tuttavia, la generalità di questa osservazione a domini più complessi rimane una questione aperta che motiva ulteriori ricerche.

Capitolo 3. L'Architettura a Incorporamento Congiunto Predittivo: Formalismo, Fondamenti e Varianti

3.1 Genesi concettuale e collocazione nell'apprendimento autosupervisionato

L'architettura a incorporamento congiunto predittiva — nota con l'acronimo JEPA (*Joint Embedding Predictive Architecture*) — nasce come proposta teorica per superare le limitazioni tanto dei modelli generativi quanto dei metodi contrastivi nell'apprendimento di rappresentazioni utili per il ragionamento e la

pianificazione. La sua formulazione originale, dovuta a LeCun nel documento programmatico del 2022 *A Path Towards Autonomous Machine Intelligence*, è notevolmente generale: la JEPA è descritta come un framework in cui un modello predittivo opera sulle rappresentazioni latenti di due osservazioni (o due viste temporalmente distanziate della stessa scena), tentando di prevedere la rappresentazione dell'una a partire dall'altra, condizionatamente a una variabile di contesto che cattura la "relazione" tra le due.

L'intuizione fondatrice è che un sistema intelligente non necessita di ricostruire fedelmente il mondo percepito: gli è sufficiente apprendere a *prevedere*, nello spazio delle rappresentazioni interne, come il proprio modello del mondo cambierà a seguito di un'azione o di uno scorrere del tempo. Questa distinzione — tra predire i pixel e predire gli embedding — ha conseguenze di vasta portata sull'architettura, sull'obiettivo di apprendimento e sulle garanzie di generalizzazione del sistema.

Per situare la JEPA nel panorama dell'apprendimento autosupervisionato, può essere utile richiamare i principali paradigmi alternativi. I metodi contrastivi — SimCLR, MoCo, BYOL — apprendono rappresentazioni massimizzando la similarità degli embedding di coppie di osservazioni correlate (viste diverse della stessa immagine) e minimizzando quella di coppie non correlate (immagini diverse). Questa strategia richiede la costruzione di una distribuzione di campioni negativi, il che introduce dipendenze dalle dimensioni del batch, dal modo in cui i negativi sono costruiti e da varie tecniche di stabilizzazione (code di negativi, momentum encoder, ecc.). I metodi di mascheramento — MAE, BEiT — apprendono rappresentazioni attraverso la previsione delle patch mascherate di un'immagine, un obiettivo generativo in spazio pixel. La JEPA si distingue da entrambi: non richiede campioni negativi espliciti, non opera nel pixel-space, e la sua struttura predittiva è naturalmente orientata alla cattura della causalità temporale piuttosto che della similarità semantica.

La differenza rispetto ai modelli generativi può essere formalizzata con precisione. Un modello generativo richiede che il decoder $d_\psi : \mathbb{Z} \to \mathcal{O}$ sia in grado di invertire, almeno approssimativamente, la mappa $enc_\theta : \mathcal{O} \to \mathbb{Z}$. Ciò implica che z_t debba conservare sufficiente informazione su o_t da consentire la ricostruzione, indipendentemente dalla sua utilità per la predizione futura. Se si misura la qualità del decoder con la perdita di ricostruzione

$$\mathcal{L}_{rec}(\theta, \psi) = \mathbb{E}_{o_t \sim \mathcal{D}}[\, \| d_\psi(enc_\theta(o_t)) - o_t \|^2_2 \,]$$

il vincolo implicito è che la mutua informazione I(o_t; z_t) sia massimizzata, ovvero che z_t sia una rappresentazione sufficientemente ricca da approssimare o_t. Più formalmente, per il teorema di data processing si ha:

$I(o_t; \hat{z}_{t+1}) \leq I(o_t; z_t) \leq H(o_t)$

La JEPA rilassa radicalmente questo vincolo: z_t deve essere sufficiente non per ricostruire o_t, bensì per predire z_{t+1} dato a_t. Le informazioni su o_t che non sono causalmente rilevanti per l'evoluzione futura possono essere scartate senza penalità. Formalmente, la JEPA mira a massimizzare la mutua informazione condizionale I(z_{t+1}; z_t, a_t) piuttosto che I(o_t; z_t).

3.2 La struttura encoder–predictor: formalismo dettagliato

L'architettura JEPA per la modellazione di ambienti dinamici si articola in due componenti parametrici distinti che operano in cascata.

Il primo è l'encoder

$enc_\theta : \mathcal{O} \rightarrow \mathbb{Z}, z_t = enc_\theta(o_t) \in \mathbb{R}^d$

il quale riceve in ingresso un'osservazione grezza $o_t \in \mathbb{R}^{H \times W \times C}$ e restituisce un vettore latente z_t di dimensione d. In LeWorldModel, l'encoder è implementato come un Vision Transformer nella configurazione *tiny* (ViT-T): 12 strati di attenzione, 3 teste per ogni strato di *multi-head self-attention*, dimensione nascosta D = 192, patch size p = 14. Per immagini 224×224, il numero di patch non sovrapposte è $(224/14)^2 = 256$ token visivi, a cui si aggiunge il token classificatore [CLS] per un totale di 257 token. L'embedding di ogni patch è calcolato da una proiezione lineare dei valori dei pixel nella patch: ogni patch 14×14×3 viene trasformata in un vettore di dimensione D = 192 mediante una matrice $W_{patch} \in \mathbb{R}^{(14 \times 14 \times 3) \times 192}$, e si aggiungono embedding posizionali appresi per codificare la posizione spaziale di ciascuna patch.

Dopo L = 12 strati di self-attention, il token [CLS] dell'ultimo strato — che aggrega informazione globale sull'intera immagine attraverso i meccanismi di attenzione — viene estratto come rappresentazione globale. Una rete di proiezione — un MLP con un singolo strato nascosto e Batch Normalization — trasforma ulteriormente questo token nella rappresentazione finale z_t. La struttura del MLP di proiezione è:

$z_t = W_2 \cdot \mathrm{BN}(\sigma(W_1 \cdot h_{[CLS]} + b_1)) + b_2$

dove $h_{[CLS]} \in \mathbb{R}^D$ è l'output del token [CLS] dell'ultimo strato del ViT, σ è una funzione di attivazione (tipicamente GELU), $W_1 \in \mathbb{R}^{D \times D}$ e $W_2 \in \mathbb{R}^{D \times d}$ sono le matrici di peso, $b_1 \in \mathbb{R}^D$ e $b_2 \in \mathbb{R}^d$ sono i bias, e BN è la Batch Normalization. La presenza della Batch Normalization è funzionalmente necessaria per due ragioni: in primo luogo normalizza la distribuzione degli input al secondo strato, stabilizzando l'addestramento; in secondo luogo, come si argomenterà nel capitolo sulla regolarizzazione SIGReg, la normalizzazione per lotto agisce implicitamente come pre-condizionamento per l'obiettivo di matching della distribuzione, consentendo a SIGReg di operare su embedding già approssimativamente centrati e con varianza controllata.

Il secondo componente è il predittore

$\mathrm{pred}_\varphi : \mathbb{Z}^N \times \mathscr{A} \to \mathbb{Z}, \hat{z}_{t+1} = \mathrm{pred}_\varphi(z_{t-N+1:t}, a_t)$

il quale riceve in ingresso una storia di N embedding latenti e l'azione corrente, e produce la previsione dell'embedding al passo successivo. In LeWorldModel, il predittore è un transformer con 6 strati, 16 teste di attenzione e dimensione nascosta $D_{pred} = 384$ (configurazione ViT-Small), per un totale di circa 10 milioni di parametri. Rispetto all'encoder ViT-T, il predittore ha più teste di attenzione — 16 contro 3 — il che gli consente di modellare relazioni più complesse tra i token della storia latente. Il mascheramento causale temporale è implementato attraverso una maschera di attenzione triangolare inferiore, che impedisce a ogni token di vedere i token successivi nella sequenza storica, garantendo che la previsione $\hat{z}_{t+1}$ dipenda solo da $z_{t-N+1}, \ldots, z_t$ e da a_t.

Le azioni vengono incorporate nel predittore tramite Adaptive Layer Normalization applicata a ogni strato. Formalmente, se $h_l \in \mathbb{R}^d$ è l'embedding nell'l-esimo strato prima della normalizzazione, AdaLN opera come:

$\mathrm{AdaLN}(h_l, a) = \gamma_l(a) \odot \mathrm{LayerNorm}(h_l) + \beta_l(a)$

dove $\odot$ è il prodotto elemento per elemento, $\gamma_l(a) \in \mathbb{R}^d$ e $\beta_l(a) \in \mathbb{R}^d$ sono rispettivamente i parametri di scalatura e traslazione condizionali all'azione, generati da un piccolo MLP applicato all'embedding dell'azione:

$[\gamma_l(a), \beta_l(a)] = \text{MLP}_l(a) \in \mathbb{R}^{2d}$

I parametri del MLP_l sono inizializzati in modo che $\gamma_l(a) = 0$ e $\beta_l(a) = 0$ per ogni a all'inizio dell'addestramento. Questa inizializzazione — ispirata al lavoro su DiT (*Diffusion Transformer*) di Peebles e Xie — garantisce che nella fase iniziale dell'addestramento il predittore operi come se non ci fosse condizionamento sull'azione, permettendo che la struttura della storia latente venga appresa prima che il condizionamento venga gradualmente introdotto. Senza questa inizializzazione, le perturbazioni casuali del condizionamento nelle prime fasi dell'addestramento potrebbero destabilizzare l'apprendimento della dinamica latente.

Anche l'uscita del predittore è seguita da una rete di proiezione analoga a quella dell'encoder — MLP con Batch Normalization — che trasforma l'embedding predetto nella stessa geometria dello spazio degli embedding target z_{t+1}. Questo garantisce la commensurabilità tra previsione e target nella perdita di predizione.

Il numero totale di parametri del modello è di circa 15 milioni: 5 milioni per l'encoder ViT-T, 10 milioni per il predittore ViT-S, più i parametri delle reti di proiezione. Questa scala modesta è compatibile con l'addestramento su una singola GPU NVIDIA L40S nell'arco di poche ore — un fattore di accessibilità che gli autori segnalano esplicitamente come motivazione della scelta architetturale.

3.3 La perdita di predizione e la sua interpretazione informazionale

L'obiettivo di apprendimento naturale per l'architettura JEPA è la minimizzazione della distanza quadratica tra l'embedding predetto e quello reale al passo successivo:

$$\mathcal{L}_{\text{pred}}(\theta, \varphi) = (1/BT) \sum_{i=1}^{B} \sum_{t=1}^{T-1} \| \hat{z}_{t+1}^{(i)} - z_{t+1}^{(i)} \|^2_2$$

dove $\hat{z}_{t+1}^{(i)} = \text{pred}_\varphi(z_{t-N+1:t}^{(i)}, a_t^{(i)})$ e $z_{t+1}^{(i)} = \text{enc}_\theta(o_{t+1}^{(i)})$, con B la dimensione del mini-batch e T la lunghezza della sotto-traiettoria.

L'addestramento avviene in modalità *teacher-forcing*: il target z_{t+1} viene sempre calcolato dall'encoder applicato all'osservazione reale o_{t+1},

piuttosto che dall'embedding predetto al passo precedente. Questa scelta stabilizza l'addestramento prevenendo l'accumulo di errori autoregressivi durante la fase di ottimizzazione — il cosiddetto *exposure bias* — pur non rispecchiando esattamente la modalità di utilizzo del modello all'inferenza, dove il predittore opera in catena aperta. La discrepanza tra la modalità di addestramento (teacher-forcing) e quella di inferenza (autoregressive rollout) è una fonte potenziale di degradazione delle prestazioni su orizzonti lunghi, e costituisce una limitazione strutturale del teacher-forcing che si manifesta nell'accumulo di errori durante la pianificazione.

L'interpretazione informazionale della perdita di predizione è illuminante. Si può riscrivere la perdita come:

$$\mathcal{L}_\text{pred}(\theta, \varphi) = \mathbb{E}_{(o_t,\ a_t,\ o_\{t+1\}) \sim \mathcal{D}} [\| \text{pred}_\varphi(\text{enc}_\theta(o_t),\ a_t) - \text{enc}_\theta(o_\{t+1\}) \|^2_2]$$

Minimizzare questa espressione richiede che l'encoder trasformi le osservazioni in rappresentazioni tali che la transizione da enc_θ(o_t) a enc_θ(o_{t+1}) condizionatamente ad a_t sia approssimabile dal predittore. Se l'encoder mappasse osservazioni con diversa struttura causalmente rilevante in embedding identici, il predittore non riuscirebbe a distinguere traiettorie diverse e non potrebbe minimizzare la perdita. La perdita incentiva quindi l'encoder a preservare le differenze tra osservazioni che conducono a evoluzioni future distinte — precisamente l'informazione causalmente rilevante per la dinamica.

Si noti tuttavia che questa incentivazione è *asimmetrica*: la perdita penalizza la mancata distinzione di stati con evoluzioni future diverse, ma non penalizza la mancata distinzione di stati con evoluzioni future identiche. Due osservazioni che differiscono solo per dettagli visivi irrilevanti per la dinamica — un'ombra, una variazione di texture, un'occlusione parziale di uno sfondo — tendono ad avere evoluzioni future simili, e l'encoder è quindi libero di codificarle in embedding vicini. Questo è precisamente il comportamento desiderato: la JEPA filtra naturalmente i dettagli visivi percettivamente rilevanti ma causalmente irrilevanti.

3.4 Istanze della famiglia JEPA: da I-JEPA a LeWorldModel

La famiglia JEPA comprende una varietà di istanze che differiscono per il dominio di applicazione, l'obiettivo di predizione e la strategia anti-collasso

adottata. Un'analisi comparativa di queste istanze è istruttiva per comprendere il posizionamento di LeWorldModel nell'evoluzione del paradigma.

I-JEPA, introdotto da Assran e collaboratori nel 2023, applica il principio JEPA all'apprendimento autosupervisionato da immagini statiche. L'encoder è un ViT applicato a un sottoinsieme di patch visibili dell'immagine, mentre il predittore ha il compito di prevedere le rappresentazioni latenti delle patch mascherate, condizionatamente alle patch visibili. La strategia anti-collasso è la coppia EMA–SG: l'encoder target — il cui output costituisce i label predetti — è aggiornato come media mobile esponenziale dell'encoder online. I-JEPA produce rappresentazioni di alta qualità per compiti di visione a valle (classificazione, segmentazione), ma richiede l'EMA e non è direttamente applicabile al dominio del controllo robotico.

V-JEPA e V-JEPA 2, estensioni al dominio video, applicano lo stesso principio a sequenze di frame. Il predittore prevede le rappresentazioni latenti dei frame mascherati condizionatamente ai frame visibili, usando architetture transformer adattate alla struttura spazio-temporale del video. La sfida specifica del video è che la predizione autoregressiva in spazio latente può essere intesa sia come previsione spaziale (completamento di porzioni mancanti di un frame) che come previsione temporale (previsione di frame futuri), e V-JEPA 2 esplora sistematicamente queste due componenti. Le strategie anti-collasso rimangono basate su EMA e SG.

PLDM, discendente diretto di questa famiglia nel dominio del controllo robotico, introduce la predizione condizionata all'azione e abbandona l'EMA in favore della regolarizzazione VICReg estesa temporalmente. Come già discusso, il prezzo è la moltiplicazione degli iperparametri e l'instabilità dell'addestramento.

DINO-WM coloca invece il problema su un piano diverso: anziché affrontare il collasso con tecniche di regolarizzazione, lo elude congelandoun encoder DINOv2 pre-addestrato. Il predittore viene addestrato su questo spazio rappresentazionale fisso, con un obiettivo puramente predittivo senza regolarizzazione.

LeWorldModel conclude questa progressione con la prima soluzione che combina l'addestramento end-to-end dalla sola perdita predittiva con una regolarizzazione formalmente garantita contro il collasso e con un singolo iperparametro efficace.

Capitolo 4. Il Collasso della Rappresentazione: Analisi Formale, Geometria dell'Ottimizzazione e Strategie Correttive

4.1 Definizione formale e geometria del collasso

Il collasso della rappresentazione è la degenerazione del sistema encoder–predittore verso una soluzione triviale in cui l'encoder mappa ogni osservazione — indipendentemente dal suo contenuto — in un vettore latente identico o quasi identico. Per comprendere pienamente il fenomeno, è utile analizzarlo da più angolazioni: la geometria degli embedding, la struttura dello spazio di ottimizzazione, e la relazione con il paesaggio della perdita.

Formalmente, il collasso si manifesta quando $\text{enc}_\theta(o) \approx c \in \mathbb{R}^d$ per ogni $o \in \mathcal{O}$, per qualche vettore costante c. In forma più generale, anche soluzioni parzialmente collassate — in cui gli embedding si concentrano su un sottospazio di dimensione $k \ll d$ — costituiscono forme di collasso che compromettono l'utilità della rappresentazione. Si può quantificare il grado di collasso attraverso la *dimensione effettiva* dell'insieme degli embedding, definita come:

$$d_{\text{eff}}(Z) = (\Sigma_i \sigma_i(Z))^2 / \Sigma_i \sigma_i(Z)^2$$

dove $\sigma_i(Z)$ sono i valori singolari della matrice degli embedding $Z \in \mathbb{R}^{B \times d}$ (dopo centratura). Questa quantità — nota come *effective rank* — misura quanto la distribuzione degli embedding è distribuita lungo le diverse direzioni dello spazio latente: vale 1 se tutti gli embedding sono allineati lungo un'unica direzione (collasso totale) e d se gli embedding sono uniformemente distribuiti lungo tutte le direzioni (piena utilità della capacità dello spazio). Una rappresentazione non collassata dovrebbe avere $d_{\text{eff}}(Z) \approx d$.

La ragione per cui la sola minimizzazione di $\mathcal{L}_{\text{pred}}$ conduce inevitabilmente al collasso è immediatamente comprensibile dalla struttura della perdita. Si consideri la soluzione costante $\text{enc}_\theta(o) = c$ per ogni o e $\text{pred}_\varphi(z, a) = c$ per ogni z, a. Sotto questa mappa, l'embedding predetto coincide sempre con l'embedding reale — entrambi uguali a c — e quindi $\mathcal{L}_{\text{pred}} = 0$. Questa è la

soluzione ottima globale della perdita di predizione pura, e qualsiasi algoritmo di discesa del gradiente tenderà a convergervi in assenza di vincoli aggiuntivi.

Per vedere questo più formalmente, si noti che la perdita di predizione può essere riscritta come:

$$\mathcal{L}_{pred}(\theta, \varphi) = \mathbb{E}[\|pred_\varphi(z_t, a_t) - z_{t+1}\|^2_2] = \mathbb{E}[\|pred_\varphi(z_t, a_t) - z_{t+1}\|^2_2]$$

Prendendo il gradiente rispetto ai parametri θ dell'encoder e annullandolo, si ottiene che il sistema può minimizzare questa perdita indifferentemente sia imparando a prevedere correttamente le transizioni latenti, sia facendo collassare tutte le osservazioni sullo stesso punto. Il paesaggio di perdita è quindi degenerato: contiene una moltitudine di minimi globali (con perdita pari a 0), alcuni dei quali corrispondono a rappresentazioni utili e molti dei quali corrispondono al collasso.

Più sottilmente, esiste anche un bias dinamico verso il collasso durante l'ottimizzazione stocastica. Le coppie di frame consecutivi (o_t, o_{t+1}) sono quasi sempre simili: il robot si muove di poco tra frame successivi, l'ambiente cambia lentamente, e le differenze visive tra frame adiacenti sono tipicamente piccole rispetto alla variabilità totale del dataset. Questo crea una pressione sistematica verso la *lentezza* delle feature — un bias verso rappresentazioni che cambiano poco nel tempo — che, non opportunamente regolata, si traduce in un encoder che ignora le differenze tra frame di lontananza temporale e finisce per codificare solo le strutture più lente della scena, degradando progressivamente verso il collasso.

4.2 Analisi della geometria del gradiente

Per comprendere la dinamica dell'addestramento in presenza di potenziale collasso, è utile analizzare la struttura dei gradienti durante l'ottimizzazione.

Il gradiente della perdita di predizione rispetto ai parametri θ dell'encoder è:

$$\partial\mathcal{L}_{pred}/\partial\theta = (2/BT)\ \Sigma_{i,t}\ (\hat{z}_{t+1}^{(i)} - z_{t+1}^{(i)}) \cdot \partial(\hat{z}_{t+1}^{(i)} - z_{t+1}^{(i)})/\partial\theta$$

Il termine $\partial\hat{z}_{t+1}^{(i)}/\partial\theta$ dipende dalla composizione $pred_\varphi \circ enc_\theta$: il gradiente si propaga dall'errore di predizione attraverso il predittore fino all'encoder dell'osservazione corrente. Il termine $-\partial z_{t+1}^{(i)}/\partial\theta$ è il

gradiente negativo che spinge l'encoder a modificare la rappresentazione di o_{t+1}^(i) in direzione del target predetto, ovvero in direzione dell'embedding predotto ẑ_{t+1}^(i).

Questo secondo termine è critico: esso spinge enc_θ(o_{t+1}) verso ẑ_{t+1} = pred_φ(enc_θ(o_t), a_t), ovvero verso ciò che il predittore si aspetta sulla base dello stato precedente. Se il predittore è già parzialmente collassato — producendo previsioni simili per stati diversi — questo gradiente spinge l'encoder a far sì che gli embedding di osservazioni diverse sembrino più simili a quanto il predittore prevede, amplificando il collasso. Si crea un circolo vizioso: il collasso del predittore induce il collasso dell'encoder, che a sua volta facilita il collasso del predittore.

Questa instabilità è di natura fondamentalmente diversa da quella osservata nei modelli generativi con perdite di ricostruzione. In un modello con decoder, il gradiente rispetto ai parametri dell'encoder include un termine che spinge l'encoder a distinguere stati diversi per consentire la ricostruzione — un segnale che contrasta direttamente il collasso. Nell'architettura JEPA pura, questo segnale è assente: l'unica pressione a mantenere la distinzione tra embedding viene dall'obiettivo di previsione, che può essere soddisfatto anche con rappresentazioni collassate.

4.3 Strategie architetturali asimmetriche: stop-gradient e medie mobili esponenziali

La prima classe di strategie anti-collasso — quella delle asimmetrie architetturali — introduce nel sistema una coppia di encoder con dinamiche di aggiornamento diverse, creando una tensione che destabilizza la soluzione triviale.

Nello schema EMA–SG, il sistema mantiene due copie dell'encoder: un encoder online enc_θ, i cui parametri vengono aggiornati con la discesa del gradiente, e un encoder target enc_{θ̄}, i cui parametri vengono aggiornati come media mobile esponenziale dei parametri online:

$$\bar{\theta} \leftarrow \tau\,\bar{\theta} + (1-\tau)\,\theta,\ \tau \in (0,1)$$

Il target z̄_{t+1} = enc_{θ̄}(o_{t+1}) viene calcolato senza propagazione del gradiente rispetto a θ̄ (stop-gradient), mentre il predittore pred_φ viene addestrato a minimizzare ‖pred_φ(z_t) − z̄_{t+1}‖². L'encoder online viene

ottimizzato solo attraverso il primo termine della differenza: $\hat{z}_{t+1} = \text{pred}_\varphi(\text{enc}_\theta(o_t), a_t)$ deve avvicinarsi a $\bar{z}_{t+1} = \text{enc}_{\bar{\theta}}(o_{t+1})$.

L'intuizione è che la lentezza dell'aggiornamento del target encoder — regolata dal parametro τ vicino a 1 — crea un "target in movimento" abbastanza stabile da fornire un segnale di addestramento utile, ma abbastanza lento da impedire che il collasso dell'encoder online si propaghi immediatamente all'encoder target. Se l'encoder online tende verso il collasso, il target encoder rimane temporaneamente più informativo, fornendo un segnale gradiente che contrasta la collassazione.

Tuttavia, la giustificazione formale di questo meccanismo rimane incompleta. Non esiste, per la coppia EMA–SG, un enunciato del tipo: "se si minimizza l'obiettivo $\mathcal{L}_{\text{EMA-SG}}$, allora il collasso è impossibile". L'argomento intuitivo descritto sopra ha carattere qualitativo e non garantisce la prevenzione del collasso per qualsiasi valore di τ, qualsiasi architettura e qualsiasi distribuzione dei dati. In particolare, si può costruire esempi in cui il sistema converge al collasso anche con EMA e SG: se il dataset è sufficientemente omogeneo — tutte le traiettorie molto simili tra loro — la pressione verso il collasso può sopraffare la resistenza dell'EMA.

Il lavoro di Ponce e collaboratori citato nell'articolo di LeWorldModel formalizza questa incompletezza teorica: dimostrano che EMA e SG non corrispondono, in generale, alla minimizzazione di alcun obiettivo ben definito, il che implica che non esiste una garanzia di convergenza a una soluzione non collassata basata sulla struttura dell'obiettivo ottimizzato.

4.4 VICReg e la regolarizzazione della struttura di covarianza

Una seconda classe di strategie, formalmente più solida, impone esplicitamente vincoli sulla struttura statistica degli embedding. VICReg, introdotto da Bardes, Ponce e LeCun, è la proposta prototipica di questa categoria.

L'idea fondamentale è che un insieme di embedding non collassato deve soddisfare tre proprietà: le diverse componenti latenti devono avere alta varianza (diversità delle feature), devono avere bassa covarianza (decorrelazione delle feature), e le rappresentazioni di osservazioni corrispondenti — sotto trasformazioni che dovrebbero essere invarianti — devono essere simili (invarianza). Questi tre vincoli danno il nome all'approccio: Variance-Invariance-Covariance regularization.

Formalmente, dati gli embedding $Z_t = \{z_t^{(i)}\}_{i=1}^B \in \mathbb{R}^{B\times d}$ al passo temporale t e la loro versione centrata $\bar{Z}_t = Z_t - (1/B)\mathbf{1}\mathbf{1}^\top Z_t$, i tre termini di perdita sono:

Termine di varianza: $\mathcal{L}_{var} = (1/Td) \Sigma_t \Sigma_d \max(0, 1 - \sqrt{(Var(z^{(d)}_t) + \varepsilon)})$

dove $Var(z^{(d)}_t) = (1/(B-1)) \Sigma_i (z^{(d,i)}_t - \mu^{(d)}_t)^2$ è la varianza empirica della d-esima componente nel batch. Il termine è nullo quando la deviazione standard di ciascuna componente supera 1 (la soglia è arbitraria ma in pratica efficace), e positivo altrimenti.

Termine di covarianza: $\mathcal{L}_{cov} = (1/d) \Sigma_{j\neq k} [Cov(Z_t)]^2_{jk}$

dove $Cov(Z_t) = (1/(B-1)) \bar{Z}_t^\top \bar{Z}_t \in \mathbb{R}^{d\times d}$. Questo termine penalizza le correlazioni tra dimensioni latenti diverse, incoraggiando l'encoder a codificare dimensioni statisticamente indipendenti. Si noti che il termine penalizza solo le entrate fuori-diagonale della matrice di covarianza, non quelle diagonali (la varianza), già gestita da $\mathcal{L}_{var}$.

Termine di invarianza (predittivo): $\mathcal{L}_{inv} = \|z_t - z'_t\|^2_2$

dove z_t e z'_t sono le rappresentazioni di due viste diverse della stessa osservazione (o, nel contesto della predizione, della stessa osservazione e della sua previsione).

L'obiettivo completo è: $\mathcal{L}_{VICReg} = \alpha \mathcal{L}_{inv} + \beta \mathcal{L}_{var} + \gamma \mathcal{L}_{cov}$

Si può mostrare che VICReg è connessa, in un senso approssimativo, alla decomposizione spettrale: minimizzare la covarianza fuori-diagonale è equivalente a spingere la matrice di covarianza degli embedding verso una matrice diagonale, il che corrisponde a un embedding che approssima i vettori propri della matrice di kernel dei dati — esattamente ciò che fanno i metodi di analisi in componenti principali (PCA) e le loro varianti non lineari. Questa connessione, esplicitata nel lavoro di Balestriero e LeCun del 2022 che mostra l'equivalenza spettrale di diversi metodi di apprendimento contrastivo e non contrastivo, suggerisce che VICReg apprende strutture globali nella distribuzione dei dati analoghe a quelle estratte da metodi di analisi spettrale classici.

PLDM estende questa architettura al caso della modellazione temporale di traiettorie, introducendo varianti temporali dei termini di varianza e covarianza. Il termine di similarità temporale:

$$\mathcal{L}_{\text{time-sim}} = (1/BT)\ \Sigma_i\ \Sigma_t\ \|z_t^{(i)} - z_{t+1}^{(i)}\|^2_2$$

incoraggia la *smoothness* delle traiettorie latenti — una preferenza per rappresentazioni che variano lentamente nel tempo, coerente con la fisica degli ambienti robotici dove i cambiamenti di stato sono continui. Il termine di varianza temporale:

$$\mathcal{L}_{\text{time-var}} = (1/Bd)\ \Sigma_i\ \Sigma_d\ \max(0,\ 1 - \sqrt{(\text{Var}(z^{(d,i)}_{1:T}) + \varepsilon)})$$

con $\text{Var}(z^{(d,i)}_{1:T}) = (1/(T-1))\ \Sigma_t\ (z^{(d,i)}_t - \mu^{(d,i)})^2$ varianza della d-esima componente lungo la traiettoria i-esima, incentiva che ogni traiettoria spazzi una porzione non trascurabile dello spazio latente — contrastando la tendenza verso traiettorie degenerate costanti. Il termine di covarianza temporale:

$$\mathcal{L}_{\text{time-cov}} = (1/B)\ \Sigma_i\ (1/d)\ \Sigma_{j\neq k}\ [\text{Cov}(Z^{(i)})]^2_{jk}$$

dove $Z^{(i)} \in \mathbb{R}^{T\times d}$ è la matrice degli embedding lungo la traiettoria i-esima e $\text{Cov}(Z^{(i)}) = (1/(T-1))\ (\tilde{Z}^{(i)})^\top\ \tilde{Z}^{(i)} \in \mathbb{R}^{d\times d}$, penalizza le correlazioni temporali tra componenti diverse. Infine, il termine della dinamica inversa $\mathcal{L}_{\text{IDM}}$ aggiunge la predizione dell'azione a_t dati z_t e z_{t+1}, fornendo un ulteriore segnale di supervisione sulla struttura della transizione.

L'obiettivo completo di PLDM è:

$$\mathcal{L}_{\text{PLDM}} = \mathcal{L}_{\text{pred}} + \alpha\ \mathcal{L}_{\text{var}} + \beta\ \mathcal{L}_{\text{cov}} + \gamma\ \mathcal{L}_{\text{time-sim}} + \zeta\ \mathcal{L}_{\text{time-var}} + \nu\ \mathcal{L}_{\text{time-cov}} + \mu\ \mathcal{L}_{\text{IDM}}$$

con sei iperparametri scalari (α, β, γ, ζ, ν, μ) da ottimizzare. La ricerca degli iperparametri ottimali in una griglia di n valori per ciascun parametro richiede n^6 valutazioni — un problema di complessità esponenziale nella dimensione del problema di ottimizzazione degli iperparametri.

Le curve di addestramento riportate nel lavoro di LeWorldModel rivelano le conseguenze di questa molteplicità: la perdita IDM di PLDM mostra oscillazioni marcate e non-monotone nel corso dell'addestramento, con valori che variano di oltre un ordine di grandezza tra iterazioni consecutive. I termini di varianza temporale e covarianza temporale mostrano comportamenti non

monotoni che suggeriscono una competizione tra forze di gradiente contrastanti. Questi sintomi di instabilità hanno conseguenze pratiche dirette: le ablazioni sulla varianza di addestramento mostrano che PLDM presenta una deviazione standard di 5.0 punti percentuali sul tasso di successo tra semi diversi, contro 2.83 di LeWorldModel e 1.63 di DINO-WM — un'evidenza quantitativa della maggiore fragilità del sistema multi-termine.

4.5 Encoder pre-addestrati congelati: la soluzione per elusione

Un approccio radicalmente diverso — e in un certo senso più pragmatico — consiste nel non affrontare il collasso direttamente, ma nell'eluderlo del tutto utilizzando un encoder pre-addestrato su larga scala e *congelato* durante l'addestramento del modello del mondo. Se enc_{θ} è un encoder pre-addestrato con parametri fissi, il sistema deve solo ottimizzare il predittore pred_φ per minimizzare:

$$\mathcal{L}_{\text{DINO-WM}}(\varphi) = (1/BT)\, \Sigma_i \Sigma_t \, \|\hat{z}_{t+1}^{(i)} - z_{t+1}^{(i)}\|^2_2$$

dove $z_{t+1}^{(i)} = \text{enc}_{\theta}(o_{t+1}^{(i)})$ è calcolato dall'encoder congelato. Poiché θ non viene aggiornato, il collasso è escluso per costruzione: la distribuzione degli embedding è quella determinata dal pre-training dell'encoder, che tipicamente mostra buona diversità per via della provenienza da addestramento su dataset eterogenei di grande scala.

DINO-WM utilizza come encoder DINOv2, un ViT addestrato con l'obiettivo DINO (*Self-DIstillation with NO labels*) su circa 124 milioni di immagini naturali. Le rappresentazioni di DINOv2 presentano proprietà notevoli: sensibilità alla struttura semantica delle immagini, invarianza a trasformazioni geometriche moderate, e una geometria dello spazio latente particolarmente favorevole per compiti di visione a valle. Il predittore di DINO-WM è addestrato sull'obiettivo $\mathcal{L}_{\text{DINO-WM}}$, senza alcuna regolarizzazione aggiuntiva degli embedding.

Il numero di token prodotti da DINOv2 per immagine è significativamente superiore a quello di un encoder compatto come ViT-T con proiezione CLS: per immagini 224×224 con patch size 14, DINOv2 produce $(224/14)^2 = 256$ token visivi, ciascuno di dimensione 768. Il predittore di DINO-WM deve quindi elaborare sequenze di 256 token per frame, contro il singolo vettore di 192 dimensioni usato da LeWorldModel. Questo differenziale — approssimativamente un fattore di $256 \times 768 / 192 \approx 1024$ in termini di dimensionalità totale dell'embedding per frame — spiega direttamente il fattore

di circa 50× nella velocità di pianificazione osservato sperimentalmente: ogni rollout del predittore di DINO-WM è circa 50 volte più lento di quello di LeWorldModel per la stessa lunghezza dell'orizzonte, poiché opera su sequenze di stati latenti di dimensionalità molto maggiore.

Il prezzo di questa soluzione è duplice. In primo luogo, la capacità rappresentazionale del sistema è limitata dalle strutture apprese dall'encoder preaddestrato su dati naturali, che possono non essere le più adatte per rappresentare la dinamica di ambienti robotici sintetici. I risultati di probing riportati nell'articolo mostrano, ad esempio, che DINOv2 ha difficoltà a codificare linearmente la posizione dell'agente in Two-Room (MSE lineare = 0.488, contro 0.008 di LeWorldModel e 0.008 di PLDM) — un compito elementare per un encoder addestrato specificamente su quel dominio. In secondo luogo, la mancanza di addestramento end-to-end significa che l'encoder non può adattarsi agli aspetti dell'ambiente che sono rilevanti per il compito specifico di controllo. Se l'ambiente robotico presenta strutture visive assenti dal dataset di pre-training — geometrie inusuali, colori non naturali, iluminazioni sintetiche — l'encoder DINOv2 può produrre rappresentazioni sub-ottimali che il predittore non riesce a sfruttare efficacemente.

4.6 Verso una teoria unificata del collasso

Prima di procedere alla descrizione di SIGReg, vale la pena abbozzare una teoria più unificata del collasso che chiarisca perché le strategie precedenti sono insufficienti e cosa renderebbe una strategia anti-collasso formalmente soddisfacente.

Il collasso è, in essenza, un problema di degenerazione della distribuzione degli embedding: la distribuzione P_Z = enc_θ#P_O (la push-forward della distribuzione delle osservazioni attraverso l'encoder) collassa verso una distribuzione concentrata. La strategia anti-collasso ideale dovrebbe:

Primo, specificare esplicitamente una distribuzione target P* che sia non concentrata e che abbia supporto sull'intero spazio latente $\mathbb{R}^d$.

Secondo, definire una divergenza $\mathscr{D}(P_Z \| P^*)$ che sia nulla se e solo se P_Z = P* e positiva altrimenti, penalizzando le distribuzioni concentrate.

Terzo, garantire che la minimizzazione della divergenza sia computazionalmente trattabile e statisticamente stabile, senza introdurre iperparametri aggiuntivi difficili da calibrare.

Quarto, garantire che l'aggiunta della divergenza all'obiettivo di predizione non crei tensioni eccessivamente complesse nel paesaggio dell'ottimizzazione.

Le strategie EMA–SG non soddisfano nessuno di questi quattro criteri formalmente. VICReg soddisfa parzialmente il terzo e il quarto, ma non specifica una distribuzione target esplicita e non garantisce formalmente l'esclusione del collasso. Gli encoder pre-addestrati soddisfano il primo in modo implicito (la distribuzione degli embedding del pre-training funge da P*) ma rinunciano all'adattabilità.

SIGReg, come si argomenterà nel prossimo capitolo, soddisfa tutti e quattro i criteri: la distribuzione target è la gaussiana isotropica standard $\mathcal{N}(0, I)$; la divergenza è la statistica di Epps-Pulley proiettata, che è nulla se e solo se le proiezioni degli embedding sono gaussiane standard (e quindi, per Cramér-Wold, P_Z = $\mathcal{N}(0, I)$); la computazione è trattabile attraverso la quadratura numerica; e la singolarità dell'iperparametro λ garantisce la semplicità del bilanciamento con l'obiettivo predittivo.

Capitolo 5. LeWorldModel: Architettura, Obiettivo di Addestramento e Regolarizzazione SIGReg

5.1 Premessa: il problema del collasso come sfida di design sistemico

Prima di addentrarsi nella descrizione tecnica di LeWorldModel, è necessario ricostruire con precisione il contesto problematico entro cui il metodo si inserisce, poiché la sua originalità è strettamente solidale con la natura e la gravità della sfida che intende risolvere. La letteratura precedente ha prodotto una serie di soluzioni parziali al problema del collasso — soluzioni che funzionano in determinati contesti ma che ciascuna introduce propri vincoli, costi o fragilità. LeWorldModel non è semplicemente un'altra soluzione parziale: è il tentativo di identificare una risposta sistematicamente più soddisfacente, che rispetti simultaneamente l'insieme completo dei vincoli desiderabili: addestramento end-to-end dai pixel, assenza di encoder pre-addestrati, assenza di euristiche architetturali come EMA e stop-gradient, un solo iperparametro effettivo, e una garanzia formale di non-collasso.

Il raggiungimento simultaneo di questi obiettivi non è banale. Come si è visto nel capitolo precedente, le strategie esistenti operano compromessi: EMA e SG

sacrificano la fondatezza formale; VICReg e PLDM sacrificano la semplicità degli iperparametri; DINO-WM sacrifica l'adattabilità dell'encoder. LeWorldModel propone una via diversa: non un compromesso tra obiettivi contrastanti, ma una soluzione che soddisfa tutti i vincoli attraverso la scelta di un regolarizzatore formalmente giustificato e computazionalmente trattabile.

5.2 Architettura del modello: encoder, predittore e rete di proiezione

La struttura di LeWorldModel si compone di due blocchi funzionali principali — encoder e predittore — la cui architettura è stata già in parte descritta nel capitolo dedicato alla famiglia JEPA. Vale tuttavia la pena di riprenderne la descrizione in modo più sistematico, evidenziando le scelte che sono funzionalmente necessarie per l'obiettivo di addestramento, quelle che sono di tipo euristico ma sperimentalmente validate, e quelle che potrebbero essere sostituite senza compromettere le proprietà fondamentali del metodo.

L'encoder enc_θ realizza la mappa

z_t = enc_θ(o_t) = Proj_encoder(ViT-T(o_t)[CLS])

dove ViT-T(o_t)[CLS] è il token classificatore dell'ultimo strato del Vision Transformer tiny, e Proj_encoder è una rete di proiezione MLP monostrato con Batch Normalization:

Proj_encoder(h) = W_out · BN(GeLU(W_in · h + b_in)) + b_out

con W_in ∈ ℝ^{192×192}, b_in ∈ ℝ^{192}, W_out ∈ ℝ^{192×d}, b_out ∈ ℝ^d, dove d = 192 nella configurazione di default.

La scelta di usare il solo token [CLS] — piuttosto che l'intera griglia di token visivi — è motivata dall'esigenza di compattezza: un singolo vettore di 192 dimensioni, anziché 256 vettori di 192 dimensioni (per immagini 224×224 con patch 14×14), riduce la dimensionalità dell'embedding di un fattore 256. Questa scelta è cruciale per la velocità di pianificazione, ma comporta una perdita di risoluzione spaziale: il token [CLS] aggrega informazione globale sull'intera immagine attraverso i meccanismi di attenzione, ma non preserva esplicitamente la struttura spaziale locale. I risultati di probing mostrano che questa perdita è accettabile per i compiti considerati — la posizione degli oggetti è recuperabile linearmente dall'embedding [CLS] con elevata precisione

— ma potrebbe diventare limitante per compiti che richiedono la localizzazione fine-grained di caratteristiche.

La presenza della Batch Normalization nella rete di proiezione è funzionalmente necessaria per l'interazione con SIGReg. L'ultimo strato del ViT applica una Layer Normalization che normalizza ogni embedding z_t a media zero e varianza unitaria *all'interno* delle componenti di ciascun vettore, ma non normalizza la distribuzione *tra* i vettori del batch. Questa differenza è cruciale: SIGReg proietta gli embedding lungo direzioni casuali e valuta la normalità delle proiezioni *nel batch*, il che richiede che la distribuzione tra gli elementi del batch sia ben comportata. La Batch Normalization, applicata alla rete di proiezione, normalizza la distribuzione degli embedding nel batch rimuovendo la media e normalizzando la varianza — operazioni che pre-condizionano la distribuzione in un modo favorevole per l'ottimizzazione di SIGReg.

Il predittore pred_φ realizza la mappa

$$\hat{z}_{t+1} = \text{pred}_\varphi(z_{t-N+1:t}, a_t) = \text{Proj_pred}(\text{ViT-S_causal}(z_{t-N+1:t}, a_t))$$

dove ViT-S_causal è il transformer con mascheramento causale e condizionamento AdaLN, e Proj_pred è una rete di proiezione analoga a quella dell'encoder. La lunghezza della storia N è impostata a 3 per Push-T e OGBench-Cube, e a 1 per Two-Room. La scelta di N = 3 per gli ambienti più complessi riflette l'utilità di informazioni contestuali su più passi temporali per la predizione della dinamica: la velocità e l'accelerazione dell'oggetto, non codificate nell'embedding istantaneo, possono essere inferite dalla differenza tra embedding consecutivi.

Il dropout del 10% nel predittore è un regolarizzatore stocastico che, come mostrato dalle ablazioni, migliora significativamente le prestazioni rispetto all'assenza totale di dropout. L'interpretazione è che il dropout impedisce al predittore di memorizzare traiettorie specifiche del dataset di addestramento, forzandolo a sviluppare una rappresentazione più robusta e generalizzabile della dinamica ambientale. Questo effetto è particolarmente rilevante per le transizioni rare ma critiche — come i movimenti bruschi o le collisioni — che potrebbero essere memorizzate senza dropout ma non generalizzate correttamente.

5.3 L'obiettivo di addestramento: derivazione e analisi delle proprietà

L'obiettivo di addestramento di LeWorldModel è la somma di due soli termini:

$$\mathcal{L}_{LeWM}(\theta, \varphi) = \mathcal{L}_{pred}(\theta, \varphi) + \lambda \cdot SIGReg(Z; \theta)$$

dove $Z = \{z_t^{(i)}\}_{i,t}$ è il tensore degli embedding latenti nel mini-batch corrente, e $\lambda > 0$ è il peso del termine di regolarizzazione. Questa binarietà non è solo un'eleganza formale: è una conseguenza strutturale della scelta di SIGReg come regolarizzatore, il quale assorbe in un singolo termine ciò che VICReg esprimeva attraverso tre termini separati (varianza, covarianza, invarianza) e PLDM attraverso sette.

5.3.1 La perdita di predizione: formalismo esteso

Il termine di predizione è la perdita MSE tra l'embedding predetto e quello reale al passo successivo:

$$\mathcal{L}_{pred}(\theta, \varphi) = (1/BT) \sum_{i=1}^{B} \sum_{t=1}^{T-1} \| pred_\varphi(enc_\theta(o_{t-N+1:t}^{(i)}), a_t^{(i)}) - enc_\theta(o_{t+1}^{(i)}) \|_2^2$$

Questa formulazione usa la notazione compatta $pred_\varphi(enc_\theta(o_{t-N+1:t}^{(i)}), a_t^{(i)})$ per indicare il predittore applicato alla storia di N embedding dell'encoder, condizionatamente all'azione. L'addestramento in teacher-forcing significa che tutti i target sono calcolati dall'encoder applicato alle osservazioni reali, non propagati autoregressivamente.

Si può riscrivere la perdita in forma matriciale per chiarirne la struttura. Sia $Z \in \mathbb{R}^{B \times T \times d}$ il tensore degli embedding e $\hat{Z} \in \mathbb{R}^{B \times T \times d}$ il tensore delle previsioni. La perdita è:

$$\mathcal{L}_{pred} = (1/BT) \| \hat{Z}_{:,1:T-1,:} - Z_{:,2:T,:} \|_F^2$$

dove $\|\cdot\|_F$ è la norma di Frobenius. Questa forma evidenzia che la perdita è una media degli errori di predizione su tutti i passi temporali e su tutti gli elementi del batch — un criterio di regressione multivariata standard, senza alcuna struttura particolare che ne faccia una perdita "speciale" nel contesto dell'apprendimento di rappresentazioni.

5.3.2 La regolarizzazione SIGReg: formalismo completo

Il termine di regolarizzazione SIGReg è il cuore della contribuzione tecnica di LeWorldModel, e merita una trattazione formale dettagliata che ne espliciti il fondamento statistico, la connessione con la teoria delle distribuzioni, e le proprietà computazionali.

Il punto di partenza è l'identificazione della distribuzione target. LeWorldModel impone che la distribuzione degli embedding Z prodotti dall'encoder converga verso la gaussiana isotropica standard in $\mathbb{R}^d$:

$$P_Z \to \mathcal{N}(0, I_d)$$

dove $0 \in \mathbb{R}^d$ è il vettore zero, $I_d \in \mathbb{R}^{d\times d}$ è la matrice identità, e la freccia indica la convergenza debole delle distribuzioni. Questa scelta è motivata da tre considerazioni. Prima, una gaussiana isotropica è la distribuzione massima-entropia con vincolo di varianza unitaria, e quindi la meno informativa a priori — il che la rende un prior ragionevole in assenza di conoscenza specifica sulla struttura dello spazio latente. Seconda, la gaussiana isotropica ha supporto su tutto $\mathbb{R}^d$, il che impedisce il collasso verso sottospazi di dimensione ridotta. Terza, la simmetria sferica della gaussiana isotropica (equivarianza rispetto alle rotazioni ortogonali) garantisce che nessuna direzione dello spazio latente sia privilegiata — una forma di equivarianza che corrisponde all'assenza di bias strutturali nella codifica.

Il problema tecnico è come valutare la distanza tra P_Z e $\mathcal{N}(0, I_d)$ in modo computazionalmente trattabile. Le divergenze classiche — Kullback-Leibler, Jensen-Shannon, distanza di Wasserstein — richiedono o la stima della densità di P_Z (computazionalmente costosa in dimensione elevata) o la soluzione di problemi di ottimizzazione complessi. SIGReg aggira questo ostacolo sfruttando due risultati matematici fondamentali.

Il *teorema di Cramér-Wold* afferma: una distribuzione di probabilità μ su $\mathbb{R}^d$ è univocamente determinata dall'insieme di tutte le sue proiezioni marginali $\{(u^\top z) : u \in S^{d-1}\}$ dove S^{d-1} è la sfera unitaria in $\mathbb{R}^d$. Formalmente: se le distribuzioni delle proiezioni univariate coincidono per ogni direzione $u \in S^{d-1}$, allora le distribuzioni multidimensionali coincidono:

$$\forall\, u \in S^{d-1}: (u^\top Z) \sim (u^\top G) \Leftrightarrow Z \sim G$$

dove $G \sim \mathcal{N}(0, I_d)$. Questo teorema riduce il problema di matching della distribuzione multidimensionale a un'infinità numerabile di problemi di matching di distribuzioni univariate.

Il vantaggio pratico è che i test di normalità univariati sono ben sviluppati, computazionalmente efficienti e statistically ben fondati. SIGReg sceglie specificamente la *statistica di test di Epps-Pulley*, basata sulla funzione caratteristica empirica (FCE).

Per un campione scalare h = (h_1, …, h_n) estratto da una distribuzione sconosciuta P_h, la FCE è:

$$\varphi_n(t; h) = (1/n) \Sigma_{k=1}^n e^{i t h_k}$$

dove i è l'unità immaginaria e $t \in \mathbb{R}$ è la frequenza. La FCE è la trasformata di Fourier della distribuzione empirica: contiene, in linea di principio, tutta l'informazione sulla distribuzione campionaria, distribuita lungo le frequenze.

La funzione caratteristica della distribuzione target — la gaussiana standard $\mathcal{N}(0,1)$ — è:

$$\varphi_0(t) = e^{-t^2/2}$$

Uno stimatore della distanza tra P_h e $\mathcal{N}(0,1)$ basato sulla FCE è l'integrale pesato della distanza quadratica:

$$T = \int_{-\infty}^{+\infty} w(t) \, |\varphi_n(t; h) - \varphi_0(t)|^2 \, dt$$

dove $w(t) \geq 0$ è una funzione di ponderazione integrabile che assegna diversi pesi alle diverse frequenze. La scelta standard è $w(t) = e^{-t^2/(2\lambda^2)}$, una gaussiana che attenuacontributi ad alta frequenza. Questa statistica — la statistica di Epps-Pulley, proposta da Thomas Epps e Lawrence Pulley nel 1983 nel contesto dei test di normalità in analisi biostatistica — può essere valutata numericamente attraverso una regola di quadratura. Con la scelta di un set di nodi $\{t_k\}_{k=1}^K$ *distribuiti uniformemente nell'intervallo [0.2, 4] e i corrispondenti pesi di quadratura* $\{w_k\}_{k=1}^K$ (regola del trapezio), si ha:

$$T \approx \Sigma_{k=1}^K w_k \, |\varphi_n(t_k; h) - e^{-t_k^2/2}|^2$$

Espandendo il modulo quadrato:

$$|\varphi_n(t; h) - e^{-t^2/2}|^2 = |\varphi_n(t; h)|^2 - 2 \, \mathrm{Re}[\varphi_n(t; h)] \, e^{-t^2/2} + e^{-t^2}$$

Sostituendo la definizione di FCE:

$|\varphi_n(t; h)|^2 = (1/n^2)\ \Sigma_{j,k}\ e^{i\,t\,(h_j - h_k)} = (1/n^2)\ \Sigma_{j,k}\ \cos(t(h_j - h_k))$

$\mathrm{Re}[\varphi_n(t; h)] = (1/n)\ \Sigma_j \cos(t\ h_j)$

La statistica di Epps-Pulley diventa quindi una funzione delle distanze tra elementi del campione $\{h_j\}$ e delle loro posizioni scalari, e può essere valutata in $O(n^2K)$ operazioni — dove n è la dimensione del campione, K il numero di nodi di quadratura — che per le dimensioni tipiche del mini-batch ($n = BT \approx 512$, $K \approx 10$) è computazionalmente trascurabile.

La regolarizzazione SIGReg applica questa valutazione a M proiezioni casuali. Siano $u^{(m)} \in S^{d-1}$, $m = 1, \ldots, M$, vettori unitari campionati uniformemente dalla sfera (nella pratica, campionando da $\mathcal{N}(0, I_d)$ e normalizzando). Le proiezioni degli embedding lungo ciascuna direzione sono:

$h^{(m)} = Z\ u^{(m)} \in \mathbb{R}^{BT}$

dove $Z \in \mathbb{R}^{BT \times d}$ è la matrice degli embedding. La statistica di Epps-Pulley calcolata su ciascuna proiezione è $T^{(m)}$. La regolarizzazione SIGReg è la media di queste statistiche:

$\mathrm{SIGReg}(Z) = (1/M)\ \Sigma_{m=1}^{M}\ T^{(m)} \geq 0$

Per il teorema di Cramér-Wold, nel limite asintotico $M \to \infty$:

$\mathrm{SIGReg}(Z) \to 0 \Leftrightarrow \forall\ u \in S^{d-1}: u^{\top}Z \sim \mathcal{N}(0,1) \Leftrightarrow P_Z \to \mathcal{N}(0, I_d)$

Questa equivalenza è il fondamento formale della garanzia di anti-collasso: SIGReg è zero se e solo se la distribuzione degli embedding è una gaussiana isotropica standard. In particolare, qualsiasi distribuzione collassata — concentrata su un punto, su una curva, su un sottospazio — produce proiezioni non gaussiane e quindi SIGReg > 0. La minimizzazione di SIGReg spinge la distribuzione degli embedding verso $\mathcal{N}(0, I_d)$, allontanandola da qualsiasi forma di collasso.

5.3.3 L'obiettivo completo e il comportamento dinamico durante l'addestramento

L'obiettivo completo di LeWorldModel è:

$\mathcal{L}_{LeWM} = \mathcal{L}_{pred} + \lambda \cdot \mathrm{SIGReg}(Z)$

con valore di default $\lambda = 0.1$ e $M = 1024$ proiezioni. Questa formulazione introduce solo due iperparametri — λ e M — ma come mostrato dalle ablazioni, M ha effetto trascurabile sulle prestazioni (il tasso di successo su Push-T rimane sostanzialmente stabile per M tra 64 e 1024), il che lascia λ come unico iperparametro effettivo. L'analisi della sensibilità a λ mostra che la finestra di robustezza è ampia: $\lambda \in [0.01, 0.2]$ produce tassi di successo superiori all'80%, con picco intorno a $\lambda \approx 0.09$. La degradazione avviene solo per $\lambda = 0.5$, dove la regolarizzazione domina l'obiettivo predittivo impedendo al modello di apprendere le strutture dinamiche.

Il comportamento dinamico dell'obiettivo durante l'addestramento è caratteristico. La perdita di predizione decresce in modo regolare e quasi monotono nel corso delle iterazioni, con un calo rapido nelle prime fasi e un plateau lento verso valori vicini a zero. La regolarizzazione SIGReg mostra invece un comportamento bifasico: una caduta brusca e rapida nelle primissime iterazioni, quando la distribuzione degli embedding prodotta dall'encoder casualmente inizializzato è lontana dalla gaussiana bersaglio, seguita da un lungo plateau in cui il regolarizzatore è sostanzialmente soddisfatto (SIGReg $\approx$ 0) e l'encoder si concentra sull'apprendimento di strutture predittive.

Questo comportamento bifasico ha un'interpretazione precisa. All'inizio dell'addestramento, i parametri dell'encoder sono inizializzati casualmente — tipicamente con una variante di inizializzazione di Xavier o He — producendo embedding con distribuzione approssimativamente gaussiana per il teorema del limite centrale (la somma pesata di molte variabili casuali tende alla gaussiana). SIGReg è quindi già parzialmente soddisfatto dall'inizializzazione, e il suo gradiente è immediatamente orientato nella giusta direzione. Man mano che l'addestramento procede e l'encoder si specializza sulla distribuzione del dataset specifico, la distribuzione degli embedding potrebbe deviare dalla gaussiana — sviluppando, ad esempio, multimodalità o asimmetrie legate alla struttura dell'ambiente — e SIGReg corregge queste deviazioni spingendo la distribuzione verso il target isotropico.

Il contrasto con le curve di addestramento di PLDM è illuminante: i sette termini di perdita di PLDM mostrano comportamenti molto diversi tra loro, con alcune componenti che crescono mentre altre decrescono, e con oscillazioni che riflettono la competizione tra i gradienti di termini con obiettivi parzialmente contrastanti. La monotonia della convergenza di LeWorldModel non è solo esteticamente più gradevole: è un indicatore di stabilità che si

traduce in riproducibilità tra semi diversi e in minore sensibilità all'ordine di presentazione dei dati durante l'addestramento.

5.4 Proprietà teoriche della regolarizzazione SIGReg

5.4.1 Il garantismo anti-collasso: enunciato formale

Si può formulare la garanzia anti-collasso di SIGReg come un enunciato matematico preciso. Sia $\mathcal{L}_LeWM(\theta, \varphi) = \mathcal{L}_pred(\theta, \varphi) + \lambda \cdot SIGReg(Z; \theta)$ l'obiettivo di addestramento. Per ogni punto stazionario (θ^*, φ^*) — ovvero ogni coppia tale che i gradienti $\partial\mathcal{L}_LeWM/\partial\theta = 0$ e $\partial\mathcal{L}_LeWM/\partial\varphi = 0$ — vale che:

$SIGReg(Z; \theta^*) = 0 \Rightarrow P_\{Z^*\} = \mathcal{N}(0, I_d)$ (per $M \to \infty$)

dove $Z^* = enc_\{\theta^*\}(\{o\^{}(i)t\}\{i,t\})$ è il tensore degli embedding all'ottimo. In altre parole, un punto stazionario con SIGReg = 0 è necessariamente un encoder che produce embedding gaussiani isotropici, escludendo per definizione il collasso.

Questo non equivale a dire che *ogni* ottimo locale dell'obiettivo è non collassato: può esistere un punto stazionario con SIGReg > 0 dove il gradiente di $\mathcal{L}_pred$ compensa esattamente il gradiente di SIGReg. Tuttavia, la penalit di SIGReg è positiva per qualsiasi distribuzione non gaussiana, e il suo gradiente punta sempre nella direzione di aumento della gaussianità — una pressione costante che rende energeticamente sfavorevoli le soluzioni collassate.

5.4.2 La scelta della gaussiana isotropica come distribuzione target: motivazioni e critiche

La scelta della gaussiana isotropica come distribuzione target merita una riflessione critica che vada oltre la semplice presentazione delle sue proprietà convenienti.

Da un lato, la gaussiana isotropica è la distribuzione massima-entropia con secondo momento fissato: tra tutte le distribuzioni con matrice di covarianza pari all'identità, $\mathcal{N}(0, I_d)$ massimizza l'entropia differenziale. Questa proprietà ha un'interpretazione informazionale precisa: imporre la gaussiana isotropica come prior sullo spazio latente equivale a imporre il minimo bias possibile sulla struttura degli embedding, dati i vincoli di varianza. In questo senso, SIGReg è

un prior minimalmente informativo, coerente con il principio di massima entropia di Jaynes.

Dall'altro, la struttura dello spazio degli stati fisico — il vero "dato" che l'encoder dovrebbe rappresentare — non è necessariamente gaussiana isotropica. Lo spazio degli stati di un ambiente robotico è tipicamente una varietà di dimensione bassa embedded in un ambiente fisico tridimensionale: le configurazioni valide del braccio robotico formano un sottoinsieme di dimensione finita dello spazio delle immagini, con struttura topologica non banale (la varietà dei possibili orientamenti di un end-effector è SO(3), il gruppo speciale ortogonale, non $\mathbb{R}^3$). Imponendo una gaussiana isotropica sull'embedding, si sta essenzialmente chiedendo all'encoder di "appiattire" questa struttura curvilinea in un embedding piatto e isotropico — un'operazione che può essere possibile senza perdita di informazione (per un'embedding dimensionalmente sufficiente di una varietà compatta, il teorema di Whitney garantisce l'esistenza di un embedding liscio in $\mathbb{R}^{2d+1}$), ma che non è necessariamente la rappresentazione più efficiente.

Gli autori riconoscono questa tensione implicitamente quando discutono i risultati sull'ambiente Two-Room. In quell'ambiente, la topologia dello spazio degli stati è non connessa — due stanze separate da un corridoio stretto — il che induce una distribuzione bimodale sulle posizioni possibili dell'agente. Il prior gaussiano isotropico impone una connessione continua tra queste due mode che non rispecchia la struttura topologica reale, potenzialmente compromettendo la qualità della rappresentazione. Più precisamente, se la distribuzione delle osservazioni presenta una struttura multimodale con modi separati da regioni a bassa probabilità, la gaussiana isotropica — unimodale per costruzione — tende a collocare embedding di osservazioni da diverse mode in regioni continue dello spazio latente, perdendo l'informazione topologica sulla separazione tra le mode.

Nonostante questa limitazione, le evidenze sperimentali suggeriscono che per la maggior parte degli ambienti robotici considerati la gaussiana isotropica è un prior sufficientemente espressivo da non impedire l'apprendimento di rappresentazioni utili. La questione aperta è se esistano distribuzioni target alternative — mixtures gaussiane, distribuzioni su varietà riemanniane, o prior appresi dai dati — che potrebbero offrire prestazioni migliori negli ambienti dove la gaussiana isotropica mostra limitazioni.

5.4.3 Connessione con la teoria dell'ottimizzazione variazionale

SIGReg può essere inquadrato nel contesto più generale della minimizzazione di divergenze tra distribuzioni — un problema che ha ricevuto crescente attenzione nell'ambito delle reti avversarie generative (GAN) e dell'inferenza variazionale. In questa prospettiva, SIGReg è equivalente a una forma approssimata di divergenza di massima discrepanza media (MMD, *Maximum Mean Discrepancy*) nella base delle funzioni caratteristiche proiettate.

Ricordando che la MMD tra due distribuzioni P e Q rispetto a un kernel κ è:

$$\mathrm{MMD}^2(P, Q; \kappa) = \mathbb{E}_{x,x'\sim P}[\kappa(x,x')] - 2\mathbb{E}_{x\sim P,\, y\sim Q}[\kappa(x,y)] + \mathbb{E}_{y,y'\sim Q}[\kappa(y,y')]$$

si può mostrare che, per opportune scelte del kernel κ (in particolare kernel gaussiani sull'argomento scalare $u^\top z$), la statistica di Epps-Pulley corrisponde a una stima della MMD tra la proiezione $(u^\top Z)$ e la distribuzione target $\mathcal{N}(0,1)$. SIGReg è quindi una media di stime di MMD lungo direzioni casuali — un estimatore della MMD che sfrutta la riduzione dimensionale per rendere la stima computazionalmente trattabile in alta dimensione.

Questa connessione con la MMD suggerisce possibili varianti di SIGReg: invece della statistica di Epps-Pulley, si potrebbe usare qualsiasi stima della MMD univariata, oppure si potrebbero usare proiezioni non uniformi (pesate per la direzione di maggiore non-gaussianità degli embedding). Queste varianti potrebbero migliorare l'efficienza statistica del regolarizzatore in termini del numero di proiezioni M necessario per una stima affidabile, ma a prezzo di maggiore complessità computazionale.

5.5 Il decoder come sonda diagnostica: implicazioni per la teoria dell'informazione della JEPA

Il decoder diagnostico — addestrato post-hoc senza alcun contributo all'obiettivo di addestramento del modello del mondo — merita un'analisi più approfondita di quanto tipicamente offerto nei lavori applicativi, poiché fornisce evidenze indirette sulla struttura informazionale dello spazio latente.

Il fatto che un decoder possa ricostruire scene visive riconoscibili a partire dal solo embedding [CLS] di dimensione 192 implica che $I(o_t; z_t)$ — la mutua informazione tra l'osservazione e il suo embedding — è significativamente positiva. Questo non contraddice il principio della JEPA: l'encoder può essere più ricco del necessario per la sola previsione. Ma pone una domanda

interessante: quanto dell'informazione preservata nell'embedding è causalmente rilevante per la previsione, e quanto è "rumore" informativo non necessario?

Si può rispondere a questa domanda attraverso il *information bottleneck* formalism. L'obiettivo IB è trovare una rappresentazione Z_t che massimizzi I(Z_t; Z_{t+1} | A_t) (informazione causalmente rilevante per la previsione) minimizzando I(O_t; Z_t) (informazione totale conservata dell'osservazione). La soluzione ottima secondo questo framework è una rappresentazione che scarta tutte le informazioni sull'osservazione che non sono predittivamente rilevanti — esattamente ciò che la JEPA pura dovrebbe apprendere.

La capacità del decoder di ricostruire le osservazioni suggerisce che LeWorldModel preserva più informazione di quanto strettamente necessario per l'obiettivo predittivo — una sorta di "ridondanza informativa" nell'embedding. Questa ridondanza non è necessariamente negativa: potrebbe riflettere strutture dell'ambiente che non sono esplicitamente richieste per la previsione a breve termine ma che sono utili per compiti a valle non previsti dall'obiettivo di addestramento. L'evidenza che l'aggiunta esplicita di una perdita di ricostruzione *degrada* le prestazioni di pianificazione (86.0% vs 96.0%) è coerente con l'ipotesi che la ricostruzione esplicita forzi l'encoder a conservare informazioni irrilevanti per il controllo — dettagli di texture, variazioni di illuminazione — che competono con le strutture causalmente rilevanti per lo spazio latente disponibile.

5.6 Analisi delle ablazioni: sensibilità dell'architettura e degli iperparametri

Le ablazioni riportate nel lavoro offrono un quadro sistematico della sensibilità di LeWorldModel rispetto alle scelte architetturali e agli iperparametri, e meritano una discussione che ne estragga le implicazioni teoriche, non solo i risultati numerici.

L'analisi sulla dimensione dell'embedding mostra un regime di saturazione: le prestazioni crescono con d fino a circa d = 192, dopo di che rimangono stabili o migliorano marginalmente fino a d = 384. Per d molto piccolo (d = 8 o d = 24), le prestazioni degradano significativamente. Questo comportamento è coerente con la teoria della capacità delle rappresentazioni: uno spazio troppo ristretto non ha sufficiente volume per contenere una gaussiana isotropica ben sviluppata senza compressioni che distruggono la struttura predittiva. Il valore di saturazione d ≈ 192 è probabilmente correlato alla dimensione intrinseca

dell'ambiente considerato — il numero di gradi di libertà fisici dello spazio degli stati — che è dell'ordine delle decine per i compiti robotici bidimensionali.

L'insensibilità alle dimensioni di SIGReg — numero di proiezioni M e numero di nodi di quadratura K — è una proprietà pratica di primo piano. Indica che la qualità della stima della distanza da $\mathcal{N}(0, I_d)$ è robusta rispetto alla risoluzione dell'approssimazione Monte Carlo e della quadratura numerica. Questo è atteso dalla teoria della MMD: la varianza dello stimatore MMD decresce come $O(1/M)$, e per $M = 64$ proiezioni la varianza è già sufficientemente bassa da non compromettere il gradiente dell'ottimizzazione.

L'ablazione sull'architettura dell'encoder — sostituzione del ViT-T con ResNet-18 — produce risultati competitivi (94.0% vs 96.0%), confermando che il metodo è robusto rispetto alla famiglia architettonica dell'encoder. Questo risultato ha implicazioni importanti per la praticità del metodo: in contesti dove i transformers sono computazionalmente proibitivi — sistemi embedded, hardware a bassa potenza — LeWorldModel potrebbe essere applicato con encoder convoluzionali senza perdita significativa di prestazioni.

Il dropout ottimale nel predittore ($p = 0.1$) è un valore moderato che fornisce sufficiente regolarizzazione senza compromettere eccessivamente la capacità del predittore. L'assenza di dropout ($p = 0.0$) produce il calo più marcato (78% vs 96%), suggerendo che senza regolarizzazione stocastica il predittore tende a mandare in overfitting le traiettorie specifiche del dataset, perdendo generalizzabilità. Valori alti di dropout ($p = 0.5$) degradano rapidamente: con metà delle unità disattivate casualmente, il predittore non ha sufficiente capacità effettiva per modellare dinamiche complesse.

L'analisi sulla dimensione del predittore rivela un pattern insolito: né ViT-T (troppo piccolo) né ViT-B (troppo grande) ottengono le prestazioni del ViT-S ottimale. Il degrado per ViT-T è atteso: capacità insufficiente per modellare le dinamiche dell'ambiente. Il degrado per ViT-B è meno ovvio e più interessante teoricamente: potrebbe riflettere una difficoltà di ottimizzazione — un modello più grande con più parametri è più difficile da addestrare con il teacher-forcing e potrebbe richiedere gradient clipping più aggressivo o schedule di apprendimento più elaborate — oppure una forma di capacità eccessiva che porta il predittore a memorizzare le transizioni del dataset piuttosto che generalizzarle.

5.7 Il fenomeno del raddrizzamento latente temporale

Uno degli aspetti più interessanti emersi dall'analisi sperimentale di LeWorldModel è il raddrizzamento spontaneo delle traiettorie latenti nel corso dell'addestramento — un fenomeno non indotto da alcun termine esplicito nell'obiettivo, ma emergente dalla combinazione dell'obiettivo predittivo e della regolarizzazione SIGReg.

Il raddrizzamento temporale si misura attraverso la *cosine similarity* tra vettori di velocità latente consecutivi. Definendo la velocità latente al tempo t come

$$v_t = z_{t+1} - z_t \in \mathbb{R}^d$$

il grado di raddrizzamento della traiettoria è:

$$S_{straight} = (1/(B(T-2))) \sum_{i=1}^{B} \sum_{t=1}^{T-2} \langle v_t^{(i)}, v_{t+1}^{(i)} \rangle / (\|v_t^{(i)}\| \cdot \|v_{t+1}^{(i)}\|)$$

Un valore S_straight vicino a 1 indica traiettorie latenti quasi rettilinee (velocità consecutive quasi parallele), mentre un valore vicino a 0 indica traiettorie con cambi di direzione frequenti.

Nel corso dell'addestramento su Push-T, S_straight per LeWorldModel cresce monotonicamente da circa 0.05 (traiettorie quasi casuali, compatibile con l'inizializzazione casuale dell'encoder) a circa 0.6 dopo 150.000 passi di addestramento. Remarkably, questo valore è significativamente superiore a quello raggiunto da PLDM (circa 0.4), nonostante PLDM includa un termine esplicito di smoothness temporale ($\mathcal{L}_{time-sim}$) mentre LeWorldModel non ha alcun termine che promuova direttamente la coerenza temporale delle traiettorie.

L'ipotesi degli autori per spiegare questo fenomeno — che SIGReg applicato *indipendentemente* a ogni passo temporale induca implicitamente una struttura lineare nelle traiettorie — è suggestiva ma merita un'analisi più approfondita. Una spiegazione alternativa è che l'obiettivo predittivo, in combinazione con la gaussianità dell'embedding, favorisca naturalmente traiettorie lineari: un predittore che deve prevedere il prossimo embedding a partire dal corrente e dall'azione troverà più facile modellare transizioni che avvengono lungo direzioni coerenti nello spazio latente — ovvero traiettorie approssimativamente lineari — rispetto a transizioni che cambiano bruscamente direzione ad ogni passo. In questo senso, il raddrizzamento

potrebbe essere interpretato come una conseguenza della pressione selettiva dell'obiettivo predittivo su un encoder che produce embedding gaussiani isotropici.

Il fenomeno del raddrizzamento latente ha connessioni con lavori in neuroscienze computazionali — in particolare con l'ipotesi del "perceptual straightening" di Hénaff, Goris e Simoncelli — che propone che il sistema visivo umano rappresenti le sequenze video naturali come traiettorie approssimativamente lineari nella propria spazio di rappresentazione. Il lavoro di Internò e colleghi ha mostrato che la misura di raddrizzamento calcolata sulle feature di DINOv2 è un discriminatore efficace per distinguere video generati da AI da video reali — suggerendo che il raddrizzamento è una proprietà della dinamica naturale che i modelli generativi faticano a riprodurre. Il fatto che LeWorldModel induca spontaneamente alti livelli di raddrizzamento è quindi una proprietà qualitativamente coerente con la fisica del mondo reale, oltre che metricamente superiore a PLDM.

5.8 Limiti strutturali e prospettive di miglioramento

Un'analisi critica di LeWorldModel non può limitarsi a descriverne le proprietà positive: deve anche esaminare onestamente i limiti strutturali e le direzioni di miglioramento che essi suggeriscono.

Il primo limite riguarda la performance sull'ambiente Two-Room, dove LeWorldModel ottiene solo l'87% di successo contro il 100% di PLDM, DINO-WM e i metodi goal-conditioned. Come discusso, la spiegazione più plausibile è che la topologia non connessa di Two-Room contrasta con il prior gaussiano isotropico di SIGReg. Una soluzione potrebbe essere l'uso di distribuzioni target più flessibili — mixtures gaussiane con un numero variabile di componenti appreso dai dati, o distribuzioni su varietà curvilinee stimate tramite tecniche di apprendimento di varietà come VAE con prior di mixture — ma a prezzo di una maggiore complessità e della perdita del singolo iperparametro.

Il secondo limite riguarda la dipendenza dall'orizzonte di pianificazione. L'accumulo di errore nel rollout autoregressivo del predittore limita l'orizzonte effettivo di pianificazione a pochi passi, costringendo l'architettura MPC a replanificare frequentemente. Per compiti che richiedono pianificazione su scale temporali lunghe — come la navigazione in ambienti estesi o le manipolazioni multi-stadio — questo limite è strutturale e non risolvibile

semplicemente aumentando H senza migliorare la qualità predittiva del modello.

Il terzo limite riguarda la necessità di osservazioni obiettivo. Il criterio di costo latente $\|\hat{z}_H - z_g\|_2^2$ richiede la disponibilità di un'immagine obiettivo o_g per ogni compito. Nei sistemi robotici reali, l'obiettivo potrebbe essere specificato in modo diverso — attraverso una descrizione linguistica, una ricompensa scalare, o una regione dello spazio degli stati — il che richiederebbe una generalizzazione del framework di pianificazione che vada oltre il semplice confronto latente.

Capitolo 6. Pianificazione Latente, il Metodo dell'Entropia Incrociata e l'Efficienza del Controllo

6.1 Il passaggio dalla rappresentazione al controllo

La distanza concettuale tra l'aver appreso un buon modello del mondo e il saper pianificare efficacemente con esso non è trascurabile. Un modello del mondo è uno strumento di *previsione*: dato uno stato corrente e un'azione, stima lo stato futuro. Un pianificatore è uno strumento di *decisione*: dato uno stato corrente e un obiettivo, seleziona la sequenza di azioni che avvicina lo stato all'obiettivo nel minor numero di passi possibile. Connettere questi due strumenti richiede un'interfaccia computazionale — una procedura di ottimizzazione che interroghi ripetutamente il modello del mondo per valutare sequenze di azioni candidate e selezionare quelle più promettenti.

In LeWorldModel questa interfaccia è il Model Predictive Control con Cross-Entropy Method (CEM), una scelta che riflette il bilanciamento tra qualità dell'ottimizzazione, efficienza computazionale e semplicità implementativa. Prima di descrivere in dettaglio l'algoritmo di pianificazione, vale la pena riflettere sui principi che lo motivano.

Il principio fondamentale è la *separazione tra apprendimento e pianificazione*. I parametri θ dell'encoder e φ del predittore, appresi nella fase offline su $\mathcal{D}$, rimangono fissi durante tutta la fase di pianificazione. L'ottimizzazione riguarda esclusivamente la sequenza di azioni $a_{1:H} \in \mathcal{A}^H$. Questa separazione ha il vantaggio di consentire la reutilizzazione dello stesso modello del mondo per compiti arbitrari, semplicemente cambiando il costo latente $\mathcal{C}$.

Ha il limite di non permettere l'adattamento online del modello alle proprietà specifiche del compito — un limite che potrebbe essere superato da approcci che fine-tunano il modello durante la pianificazione, ma a prezzo di una maggiore complessità computazionale.

6.2 Formulazione del problema di controllo ottimale in spazio latente

Il problema di pianificazione è formalizzato come un problema di controllo ottimale a orizzonte finito in spazio latente. Dato l'embedding iniziale z_1 = enc_θ(o_1) e quello obiettivo z_g = enc_θ(o_g), si cerca la sequenza di azioni ottimale:

$$a^{*}_{1:H} = \operatorname{argmin}_{a_{1:H} \in \mathcal{A}^H} \mathcal{C}(\hat{z}_H)$$

soggetta alla dinamica latente:

$$\hat{z}_1 = z_1 \quad \hat{z}_{t+1} = \text{pred}_\varphi(\hat{z}_t, a_t),\ t = 1, \ldots, H-1$$

e al costo terminale:

$$\mathcal{C}(\hat{z}_H) = \|\hat{z}_H - z_g\|_2^2$$

Questa formulazione utilizza un costo puramente terminale — definito solo sull'ultimo stato del rollout — anziché un costo cumulativo lungo tutta la traiettoria. La scelta è motivata dall'esigenza di semplicità e dall'esperienza empirica che mostra come il costo terminale sia sufficiente per il raggiungimento dell'obiettivo nei contesti considerati. In linea di principio, si potrebbe aggiungere un termine di costo intermedio:

$$\mathcal{C}_{\text{total}}(\hat{z}_{1:H}) = \sum_{t=1}^{H} \gamma^{H-t} \|\hat{z}_t - z_g\|_2^2$$

dove $\gamma \in (0,1]$ è un fattore di sconto che bilancia il costo dell'obiettivo a diversi orizzonti. Per $\gamma = 0$ si ottiene il costo puramente terminale; per $\gamma = 1$ si ottiene un costo uniforme lungo tutta la traiettoria. L'aggiunta di un termine di costo intermedio potrebbe migliorare la qualità del piano su compiti dove il raggiungimento dell'obiettivo richiede il transito attraverso stati intermedi specifici, ma introdurrebbe un ulteriore iperparametro γ da calibrare.

Una variante più sofisticata è il costo basato sulla distanza latente con pesi anisotropici:

$\mathscr{C}$_M(ẑ_H) = (ẑ_H − z_g)^⊤ Σ^{-1} (ẑ_H − z_g)

dove Σ è la matrice di covarianza empirica degli embedding nel dataset. Questa variante compensa le eventuali anisotropie dello spazio latente — direzioni lungo cui la varianza degli embedding è maggiore o minore — pesando diversamente le deviazioni in direzioni diverse. Poiché SIGReg tende a isotropizzare la distribuzione degli embedding, Σ ≈ I_d nella pratica, e le due forme di costo sono approssimativamente equivalenti.

6.3 Il Metodo dell'Entropia Incrociata: derivazione formale e proprietà

Il Metodo dell'Entropia Incrociata (CEM) è un algoritmo di ottimizzazione stocastica proposto originariamente da Rubinstein e Kroese per problemi di ottimizzazione combinatoria, e successivamente adattato per l'ottimizzazione continua nel contesto del controllo basato su modelli.

L'idea fondamentale può essere derivata dal principio di minima divergenza di Kullback-Leibler. Si consideri il problema di trovare la distribuzione π* sulle sequenze di azioni che minimizzi il valore atteso del costo:

π* = argmin_{π ∈ Δ($\mathscr{A}$^H)} 𝔼_{a_{1:H} ~ π}[$\mathscr{C}$(ẑ_H)]

dove Δ($\mathscr{A}$^H) è lo spazio delle distribuzioni di probabilità su $\mathscr{A}$^H. La soluzione esatta di questo problema richiede, in generale, la valutazione dell'intera distribuzione degli costi su $\mathscr{A}$^H, il che è computazionalmente proibitivo. Il CEM approssima iterativamente π* restringendo la ricerca a una famiglia parametrica di distribuzioni — tipicamente gaussiane — e aggiornando i parametri basandosi sulle osservazioni campionarie.

Formalmente, il CEM cerca la distribuzione gaussiana π_θ = $\mathscr{N}$(μ, Σ) che minimizzi la divergenza KL dalla distribuzione concentrata attorno ai minimi del costo:

min_{μ, Σ} KL(π_{target} ‖ π_{μ,Σ})

dove π_{target} ∝ exp(−$\mathscr{C}$(ẑ_H)/τ) è la distribuzione di Gibbs con temperatura τ → 0 che concentra la massa sulle sequenze di azioni con costo minimo. Nel limite τ → 0, π_{target} converge alla distribuzione di Dirac centrata sull'ottimo — esattamente ciò che si vuole trovare. Per τ finito, la soluzione è la gaussiana con media e covarianza corrispondenti ai momenti di

π_{target} — ovvero la gaussiana con la stessa media e varianza delle sequenze di azioni ponderate con peso proporzionale a $\exp(-\mathscr{C}/\tau)$.

L'approssimazione operativa del CEM sostituisce la media pesata rispetto a π_{target} con una media sulle K élite — le K sequenze con costo più basso tra le N campionate — il che corrisponde al limite $\tau \to 0$ con la distribuzione dei K quantile inferiori del costo:

$\mu_{t+1} = (1/K) \Sigma_{i \in \mathscr{E}_t} a^{(i)}_{1:H}$ $\Sigma_{t+1} = (1/(K-1)) \Sigma_{i \in \mathscr{E}_t} (a^{(i)}_{1:H} - \mu_{t+1})(a^{(i)}_{1:H} - \mu_{t+1})^\top$

dove $\mathscr{E}_t = \{i : \mathscr{C}(\hat{z}_H^{(i)})$ è tra i K più bassi$\}$. Nella pratica si usa spesso una struttura diagonale per Σ — ovvero si stimano solo le varianze marginali, non le covarianze — per motivi di semplicità computazionale e di stabilità statistica della stima con K campioni.

Nella configurazione di LeWorldModel: N = 300 sequenze candidate per iterazione, K = 30 élite (10% delle sequenze), T = 30 iterazioni per Push-T e T = 10 per gli altri ambienti, orizzonte H = 5 passi. Questo porta a un totale di $300 \times 30 = 9.000$ (Push-T) o $300 \times 10 = 3.000$ (altri ambienti) valutazioni del modello del mondo per ogni ciclo di pianificazione, dove ogni valutazione consiste in un rollout autoregressivo di H = 5 passi del predittore. Con embeddings di dimensione 192 e un predittore di 10M parametri, ogni rollout richiede 5 passaggi in avanti nel transformer — computazionalmente leggeri grazie alla compattezza degli embedding — portando al tempo di pianificazione totale di circa 0.98 secondi misurato empiricamente.

Il confronto con DINO-WM è istruttivo. Il predittore di DINO-WM deve elaborare sequenze di 256 token di dimensione 768 per ogni frame, il che porta a un costo computazionale per rollout circa $256 \times 768 / 192 \approx 1024$ volte superiore a quello di LeWorldModel. Con le stesse $300 \times 30 = 9.000$ valutazioni, il tempo totale di pianificazione scala a circa $0.98 \times 1024 /$ scale_factor $\approx$ 47 secondi, coerente con i 47 secondi riportati sperimentalmente.

Il differenziale di velocità di circa 48× ha implicazioni dirette sulla qualità della pianificazione sotto budget computazionale fisso. Se il budget è limitato a, diciamo, 1 secondo di tempo di calcolo, LeWorldModel può eseguire un ciclo completo di ottimizzazione CEM con 300×30 iterazioni, mentre DINO-WM può eseguire al più $300 \times 30/48 \approx 190$ valutazioni — equivalente a circa una iterazione del CEM su 190 sequenze candidate. Con una sola iterazione del

CEM, il piano ottimizzato è di qualità notevolmente inferiore rispetto a quello prodotto da 30 iterazioni, il che spiega il crollo del tasso di successo di DINO-WM da circa 90% (senza vincolo di FLOPs) a 13% (con FLOPs fissi pari a quelli di un ciclo LeWorldModel).

6.4 La strategia MPC con orizzonte scorrevole: razionale e implementazione

Il piano ottimizzato tramite CEM non viene eseguito integralmente nell'ambiente reale. Viene invece adottata una strategia di controllo predittivo a orizzonte scorrevole (receding horizon MPC): dopo ogni ciclo di ottimizzazione, l'intera sequenza ottimizzata di $H = 5$ azioni viene eseguita nell'ambiente, dopodiché il sistema acquisisce l'osservazione aggiornata e ripianifica da capo.

In LeWorldModel, $K = H = 5$, ovvero tutte le azioni del piano vengono eseguite prima della riplanning. Questa scelta differisce dalla versione più conservativa dell'MPC in cui si esegue solo la prima azione ($K = 1$) e poi si ripianifica immediatamente — una scelta che massimizza la frequenza di correzione del piano ma moltiplica il tempo di pianificazione per il numero di passi dell'esecuzione. La scelta $K = H = 5$ è un compromesso pragmatico: con un orizzonte di 5 passi e un tempo di pianificazione di circa 1 secondo, eseguire tutte le 5 azioni prima di ripianificare porta a un tempo di controllo effettivo di circa 0.2 secondi per passo — un ritardo accettabile per ambienti robotici con dinamiche non troppo rapide.

La razionale del receding horizon MPC può essere formalizzata in termini di errori di predizione. Sia $\varepsilon_t = z_t - \hat{z}_t$ l'errore di predizione al passo t, dove $z_t = \text{enc}_\theta(o_t)$ è l'embedding dell'osservazione reale e $\hat{z}_t$ è la previsione autoregressiva a partire dall'ultimo passo di replanning. Assumendo un modello lineare approssimato $\text{pred}_\varphi(z, a) \approx A\,z + B\,a$ per semplicità analitica, l'errore si accumula come:

$$\varepsilon_t = z_t - \hat{z}_t = (A + \delta A)\, z_{t-1} + B\, a_{t-1} + \text{noise} - A\, \hat{z}_{t-1} - B\, a_{t-1} = A\, \varepsilon_{t-1} + \delta A\, z_{t-1} + \text{noise}$$

dove δA è l'errore del modello (differenza tra la vera matrice di transizione e quella appresa) e noise è il rumore di processo. Nella norma, $\|\varepsilon_t\| \approx \|A\|\, \|\varepsilon_{t-1}\| + \|\delta A\|\, \|z_{t-1}\| + \|\text{noise}\|$, il che mostra che l'errore cresce (approssimativamente) geometricamente con il numero di passi dall'ultimo

replanning. La frequenza di replanning deve essere sufficientemente alta da mantenere $\|\varepsilon_t\|$ sotto una soglia di qualità, ma sufficientemente bassa da consentire al pianificatore di completare il ciclo di ottimizzazione prima che l'errore diventi incontrollabile.

Per orizzonti H = 5 passi, l'errore accumulato nelle configurazioni sperimentali è empiricamente accettabile: i rollout del predittore visualizzati nell'articolo mostrano una buona corrispondenza tra traiettorie previste e reali per T = 15-25 frame (3-5 passi di replanning), con discrepanze progressivamente crescenti a orizzonti più lunghi — particolarmente evidenti per la posizione dell'end-effector in OGBench-Cube, un ambiente 3D con dinamiche più complesse.

6.5 Analisi delle prestazioni: confronto sistematico tra metodi

Il confronto sperimentale è condotto su quattro ambienti con caratteristiche fisiche e di complessità visiva molto diverse, il che consente di valutare la robustezza dei diversi metodi rispetto alla varietà del dominio.

Su Push-T — un ambiente 2D di manipolazione dove l'agente (punto blu) deve spingere un blocco a forma di T verso una configurazione obiettivo — LeWorldModel raggiunge il 96% di successo su 50 traiettorie di valutazione, superando sia PLDM (78%) che DINO-WM senza propriocezione (74%) e persino DINO-WM con input propriocettivi aggiuntivi (92%). La superiorità rispetto a DINO-WM con propriocezione è particolarmente significativa: indica che LeWorldModel, operando esclusivamente sui pixel senza accesso a informazioni di stato privilegiate come la posizione dei giunti, riesce a estrarre dall'immagine strutture latenti più utili per la pianificazione di quanto DINO-WM riesca a fare combinando informazioni visive e propriocettive. I risultati di probing (Tab. 1 dell'articolo) confermano questa interpretazione: l'embedding di LeWorldModel codifica la posizione dell'agente con MSE lineare di 0.052 (contro 1.888 di DINO-WM e 0.090 di PLDM) e la posizione del blocco con MSE lineare di 0.029 (contro 0.006 di DINO-WM e 0.122 di PLDM) — un quadro di vantaggio relativo che varia per quantità fisica e metodo.

Su Reacher — un braccio a due giunti che deve raggiungere una configurazione obiettiva in un piano 2D — LeWorldModel ottiene l'86% contro il 78% di PLDM e il 79% di DINO-WM. Il vantaggio è più contenuto ma consistente.

Su OGBench-Cube — un ambiente 3D di manipolazione robotica con un end-effector che deve posizionare un cubo — il quadro si inverte parzialmente: DINO-WM (86%) supera LeWorldModel (74%), mentre PLDM (65%) rimane inferiore. Gli autori attribuiscono questo risultato alla maggiore complessità visiva dell'ambiente 3D e alla varietà degli stati dell'end-effector, che rendono più difficile l'addestramento di un encoder end-to-end. In questo contesto, le rappresentazioni ricche appreseche da DINOv2 su 124M immagini naturali offrono un vantaggio che il dataset limitato di 10.000 episodi di OGBench-Cube non consente di colmare con l'addestramento specifico del dominio.

Notevolmente, i metodi goal-conditioned di rinforzo offline — GCIVL (84%) e GCIQL (64%) — mostrano prestazioni competitive su OGBench-Cube, con GCIVL addirittura superiore a LeWorldModel. Questo suggerisce che l'informazione implicita contenuta nella struttura del dataset — le traiettorie eseguite da una politica pseudo-esperta che riesce a risolvere il compito — è sufficientemente ricca da consentire l'apprendimento di politiche efficaci attraverso l'imitazione comportamentale e l'apprendimento per rinforzo offline, senza necessità di un modello del mondo esplicito. Per questo tipo di ambienti complessi, il vantaggio dei modelli del mondo latenti — la generalizzabilità a nuovi compiti senza riaddestrare — potrebbe essere più rilevante del vantaggio assoluto di prestazioni.

Su Two-Room — un ambiente di navigazione 2D con due stanze connesse da un passaggio — LeWorldModel (87%) è superato da quasi tutti gli altri metodi: GCBC (100%), GCIQL (100%), GCIVL (100%), PLDM (97%), DINO-WM (100%). Come discusso in precedenza, la spiegazione più probabile è la combinazione della topologia non connessa dell'ambiente con il prior gaussiano isotropico di SIGReg. Ma vale la pena notare anche un'interpretazione alternativa: Two-Room è l'ambiente con la più bassa diversità dinamica del dataset — l'agente naviga sempre dalla stessa parte all'altra attraverso lo stesso passaggio — il che rende le transizioni ambientali particolarmente prevedibili. In questo regime di alta prevedibilità, il vantaggio predittivo della JEPA rispetto ai metodi di imitazione comportamentale diretta è minimo: la politica ottimale può essere appresa quasi direttamente per imitazione senza necessità di un modello del mondo.

6.6 Il framework di valutazione: metodologia e criteri

La metodologia sperimentale adottata per la valutazione merita un'analisi che vada oltre la semplice descrizione dei protocolli, per esaminare in che misura le scelte metodologiche influenzano l'interpretazione dei risultati.

Il protocollo di campionamento degli obiettivi — lo stato obiettivo viene scelto come lo stato che occorre nella stessa traiettoria del dataset un certo numero fisso di passi più avanti — garantisce la raggiungibilità dell'obiettivo (poiché la politica comportamentale ha dimostrato che quella traiettoria è possibile) e la coerenza con la distribuzione delle dinamiche nel dataset. Questa scelta è conservativa: in un sistema reale, gli obiettivi potrebbero essere specifici da una distribuzione diversa da quella delle traiettorie del dataset, il che potrebbe compromettere la generalizzabilità dei risultati. D'altro canto, è metodologicamente prudente: confrontare metodi su obiettivi fuori-distribuzione richiederebbe una valutazione separata della capacità di generalizzazione, complicando l'interpretazione.

La scelta di mantenere gli iperparametri fissi attraverso tutti gli ambienti — anziché ottimizzarli per-ambiente come fa PLDM nel suo articolo originale — è una scelta di conservatorismo metodologico che favorisce la valutazione della robustezza su quella delle prestazioni assolute. Un metodo che richiede calibrazione estesa per ogni ambiente è meno pratico di uno che funziona bene "out-of-the-box" con gli stessi iperparametri, anche se le prestazioni assolute dell'approccio calibrato possono essere superiori. In questo senso, i risultati di LeWorldModel sono conservativi: è possibile che un'ulteriore calibrazione di λ per ogni ambiente produca risultati migliori, anche se le ablazioni suggeriscono che i miglioramenti sarebbero marginali.

Capitolo 7. Comprensione Fisica Latente: Probing, Decodifica e Violazione dell'Aspettativa

7.1 La questione della comprensione fisica nelle rappresentazioni latenti

Una domanda fondamentale che va al di là delle prestazioni di pianificazione è: le rappresentazioni latenti apprese da LeWorldModel codificano strutture fisiche significative, o sono semplici pattern predittivi senza interpretabilità fisica? La risposta a questa domanda ha implicazioni sia scientifiche — per la

comprensione di cosa apprendono le reti neurali nei contesti di modellazione dinamica — sia pratiche — per la possibilità di diagnosticare, interpretare e migliorare questi sistemi.

LeWorldModel affronta questa domanda attraverso due paradigmi sperimentali complementari: il *probing* delle quantità fisiche e la valutazione della *violazione dell'aspettativa*. Il primo misura quanto le quantità fisiche dell'ambiente sono linearmente o non-linearmente accessibili dallo spazio latente; il secondo misura quanto il modello del mondo è "sorpreso" da eventi che violano la fisica appresa.

7.2 Probing delle quantità fisiche: metodo e risultati

Il probing delle rappresentazioni è una tecnica standard nella letteratura sull'interpretabilità delle reti neurali: si addestrano classificatori o regressori ("sonde") che prendono in input gli embedding della rete e cercano di predire proprietà fisiche note dell'ambiente. Se la sonda lineare riesce a predire la proprietà con alta accuratezza, si conclude che l'informazione è *linearmente accessibile* nell'embedding — ovvero che esiste una direzione nello spazio latente lungo cui la variazione corrisponde alla variazione della proprietà fisica. Se solo la sonda non-lineare (MLP) riesce, l'informazione è presente ma *intrecciata* con altre strutture, richiedendo una trasformazione non-lineare per essere estratta.

In LeWorldModel, le sonde sono addestrate su un dataset separato di incorporamenti con le corrispondenti annotazioni fisiche — posizioni, angoli, configurazioni articolari — generate dallo stesso ambiente di simulazione. Per valutare l'accessibilità lineare, si usa una regressione lineare (LSE). Per l'accessibilità non-lineare, si usa un MLP con due strati nascosti.

I risultati su Push-T (Tab. 1 dell'articolo) mostrano che LeWorldModel supera PLDM su tutte le quantità fisiche sia nella sonda lineare che in quella non-lineare. Per la posizione dell'agente, il coefficiente di Pearson della sonda lineare è r = 0.974 (contro 0.955 di PLDM), e la sonda MLP raggiunge r = 0.998 (contro 0.993). Per la posizione del blocco, r lineare = 0.986 (contro 0.938 di PLDM) e r MLP = 0.999 (contro 0.994). Per l'angolo del blocco — una quantità fisica più difficile da codificare per la sua natura angolare — r lineare = 0.902 (contro 0.745 di PLDM) e r MLP = 0.990 (contro 0.972). In tutti i casi, le sonde non-lineari producono correlazioni quasi perfette, suggerendo che tutta l'informazione fisica rilevante è presente nell'embedding, ma non necessariamente in forma linearmente accessibile.

DINO-WM mostra prestazioni di probing eccezionalmente alte sulla posizione del blocco (r lineare = 0.997) ma inferiori a LeWorldModel sulla posizione dell'agente (r lineare = 0.977 vs 0.974). Questa disparità è attribuita al fatto che DINOv2 è stato addestrato su 124M immagini naturali che includono molte scene con oggetti simili ai blocchi, ma non con punti agenti simili al punto blu di Push-T.

I risultati su OGBench-Cube (Tab. 4 dell'articolo) rivelano un quadro più sfumato. Per le quantità posizionali — posizione del cubo, posizione dell'end-effector — LeWorldModel uguaglia o supera PLDM: r MLP = 0.999 per la posizione del cubo (contro 0.999 di PLDM) e r MLP = 0.998 per la posizione dell'end-effector (contro 0.993). Per le quantità rotazionali — orientamento del cubo (quaternione e yaw), orientamento dell'end-effector (yaw) — tutti i metodi mostrano prestazioni di probing scarse, con r lineare ≈ 0.1−0.3. Questo suggerisce che le rotazioni 3D sono strutturalmente difficili da codificare in uno spazio latente di dimensione moderata per qualsiasi metodo, non solo per LeWorldModel. La geometria delle rotazioni SO(3) è non-euclidea e non lineare, il che rende difficile per una sonda lineare estrarne l'informazione anche se essa è presente nell'embedding.

7.3 Il paradigma della violazione dell'aspettativa: fondamenti e implementazione

Il secondo strumento di analisi della comprensione fisica è il paradigma della *violazione dell'aspettativa* (VoE), ispirato alla metodologia usata in psicologia dello sviluppo per studiare la comprensione della fisica negli infanti e recentemente adottato nell'apprendimento automatico. L'idea di fondo è che un sistema con buona comprensione della fisica debba essere "sorpreso" — ovvero assegnare alta incertezza predittiva — quando osserva eventi che violano le leggi fisiche che ha appreso, e bassa sorpresa per eventi fisicamente plausibili.

In LeWorldModel, la sorpresa è misurata quantitativamente come l'errore quadratico di predizione del modello del mondo:

$$\text{Surprise}(t) = \| \text{pred}_\varphi(\text{enc}_\theta(o_{t-1}), a_{t-1}) - \text{enc}_\theta(o_t) \|^2_2$$

ovvero la distanza nello spazio latente tra la previsione del modello e l'embedding dell'osservazione effettivamente ricevuta. Quando l'evento osservato è fisicamente plausibile e coerente con le dinamiche apprese, questa distanza dovrebbe essere piccola. Quando l'evento viola le aspettative fisiche

del modello — perché è fisicamente impossibile o estremamente improbabile data la storia osservata — la distanza dovrebbe essere grande.

Gli esperimenti di VoE confrontano tre tipi di traiettorie: una traiettoria non perturbata (baseline), una traiettoria con perturbazione visiva (il colore di un oggetto cambia improvvisamente durante l'episodio) e una traiettoria con perturbazione fisica (un oggetto viene teletrasportato a una posizione casuale, violando la continuità fisica).

Per tutti e tre gli ambienti testati — Two-Room, Push-T e OGBench-Cube — la perturbazione fisica (teletrasporto) produce un picco marcato nella sorpresa nel momento in cui avviene (t_perturb), con valori significativamente superiori alla baseline non perturbata (test t appaiato, $p < 0.01$). La perturbazione visiva (cambio di colore) produce un effetto molto più contenuto e non significativo per tutti gli ambienti eccetto Two-Room: il modello è sensibile principalmente alle perturbazioni fisiche piuttosto che a quelle visive — esattamente il comportamento desiderato da un modello del mondo che deve catturare la fisica, non la fotometria.

Questo risultato è coerente con la tesi che l'obiettivo predittivo di LeWorldModel, combinato con la regolarizzazione SIGReg, induce il sistema a codificare prioritariamente le strutture fisicamente rilevanti — posizioni, momenti, forze — piuttosto che i dettagli visivi superficiali come il colore. Un modello che avesse memorizzato le transizioni del dataset sulla base di feature visive superficiali — come un modello generativo che apprende le correlazioni tra pixel — sarebbe ugualmente sorpreso dal cambio di colore e dal teletrasporto, poiché entrambi producono variazioni visive inusuali. Il fatto che LeWorldModel sia selettivamente sensibile al teletrasporto ma non (o molto meno) al cambio di colore indica una forma di astrazione fisica nella rappresentazione.

7.4 Il confronto con PLDM e DINO-WM nella valutazione VoE

Il confronto dei grafici di sorpresa tra LeWorldModel, PLDM e DINO-WM rivela differenze qualitative interessanti nella struttura della comprensione fisica delle diverse architetture.

Per PLDM (Fig. 13 dell'articolo), la sorpresa per la perturbazione visiva è più alta e più comparabile alla sorpresa per la perturbazione fisica rispetto a LeWorldModel, specialmente in Two-Room e Push-T. Questo suggerisce che le

rappresentazioni di PLDM sono più sensibili ai cambiamenti visivi superficiali — probabilmente a causa dei termini VICReg che spingono le rappresentazioni a preservare la varianza in tutte le dimensioni, incluse quelle che codificano dettagli visivi non fisici.

Per DINO-WM (Fig. 14 dell'articolo), il quadro è diverso: la scala assoluta degli errori di predizione è molto più grande (MSE dell'ordine di 200-400 contro 5-20 per LeWorldModel e PLDM), il che riflette la diversa natura degli embedding di DINOv2 — vettori di token spaziali di dimensione 768 invece di singoli vettori globali di dimensione 192. La struttura relativa delle sorprese è simile a quella di LeWorldModel per Two-Room e Push-T, ma per OGBench-Cube DINO-WM non mostra un aumento significativo della sorpresa per nessuno dei due tipi di perturbazione — un risultato che potrebbe indicare una minore capacità di distinguere stati fisicamente anomali in ambienti 3D complessi.

7.5 La visualizzazione dello spazio latente: struttura geometrica e proprietà di separazione

La visualizzazione dello spazio latente attraverso t-SNE — una tecnica di riduzione dimensionale non-lineare — offre un'ulteriore finestra sulla struttura geometrica delle rappresentazioni apprese.

Per Push-T, la griglia di stati fisici (spostamento dell'agente e del blocco nel piano x-y) mostra, nello spazio latente bidimensionale ridotto con t-SNE, una struttura che preserva le relazioni di vicinanza: stati fisicamente vicini (stesso blocco, agente in posizioni adiacenti) appaiono vicini nel t-SNE, e la struttura spaziale bidimensionale dell'ambiente è chiaramente riconoscibile nella topologia del grafico. Questa proprietà — la preservazione locale delle relazioni di vicinanza fisica — è un indicatore positivo della qualità dell'embedding: non solo le quantità fisiche sono predittivamente utili, ma la geometria dello spazio latente rispecchia la geometria fisica dell'ambiente.

Vale la pena notare che la preservazione della struttura geometrica non è garantita dall'obiettivo di addestramento di LeWorldModel, né è direttamente incentivata da SIGReg. È un fenomeno emergente: la struttura fisica dello spazio degli stati è indotta implicitamente dalla coerenza temporale delle traiettorie — osservazioni fisicamente vicine tendono a essere temporalmente adiacenti nelle traiettorie, e l'obiettivo predittivo penalizza le mappe che spezzano questa coerenza. In questo senso, la geometria dello spazio latente

riflette la topologia delle traiettorie nel dataset piuttosto che una proprietà intrinseca dell'architettura.

7.6 Implicazioni per la riusabilità e il trasferimento delle rappresentazioni

L'insieme delle evidenze raccolte attraverso probing, decodifica e VoE suggerisce una caratterizzazione più precisa di cosa LeWorldModel apprende. Lo spazio latente non è semplicemente uno spazio di previsione — un insieme di vettori utili per calcolare il passo successivo — ma una rappresentazione fisica significativa che codifica posizioni, configurazioni e dinamiche in modo strutturato e interpretabile.

Questa caratterizzazione ha implicazioni importanti per il trasferimento delle rappresentazioni a compiti nuovi. Se lo spazio latente codifica quantità fisiche generali — posizioni, velocità, forze — piuttosto che strutture specifiche per il compito di addestramento, allora le stesse rappresentazioni potrebbero essere utili per una varietà di compiti che richiedono la manipolazione di quelle quantità. La fine-tuning dell'encoder su nuovi compiti potrebbe partire da una rappresentazione già fisicamente strutturata, riducendo il numero di campioni necessari per raggiungere buone prestazioni — un fenomeno analogo al *transfer learning* nei modelli di visione pre-addestrati, ma qui con le garanzie formali di SIGReg piuttosto che con la dipendenza da un vasto dataset di pre-training.

Capitolo 8. Conclusioni: Contributi, Limiti e Prospettive

8.1 Sintesi dei contributi tecnici

LeWorldModel rappresenta un punto di sintesi metodologicamente significativo nell'evoluzione delle architetture JEPA per il controllo robotico. I suoi contributi tecnici principali possono essere sintetizzati in tre enunciati formali:

Primo, la dimostrazione che è possibile addestrare end-to-end un'architettura JEPA per la modellazione del mondo direttamente da pixel, senza encoder pre-addestrati, senza stop-gradient e senza medie mobili esponenziali,

ottenendo prestazioni competitive o superiori ai metodi dello stato dell'arte su un insieme eterogeneo di ambienti 2D e 3D.

Secondo, la formalizzazione di SIGReg come regolarizzatore di distribuzione con garanzie provabili di anti-collasso, derivate dal teorema di Cramér-Wold e dalla proprietà di caratterizzazione univoca delle distribuzioni attraverso le loro proiezioni marginali. Questa garanzia è più forte di qualsiasi altra offerta dai metodi precedenti nell'ambito della famiglia JEPA applicata al controllo.

Terzo, la dimostrazione che la riduzione da sei a un iperparametro effettivo — conseguenza della scelta di SIGReg in luogo di VICReg esteso — porta a un addestramento più stabile, a una minore varianza tra repliche indipendenti, e a una ricerca degli iperparametri con complessità logaritmica anziché polinomiale di alto grado.

8.2 Posizionamento rispetto alla tradizione della ricerca

Dal punto di vista della tradizione della ricerca in apprendimento autosupervisionato, LeWorldModel si colloca all'intersezione di due filoni che raramente si incontrano in modo così diretto: la teoria statistica del matching di distribuzioni — con le sue connessioni con la MMD, i test di normalità multivariata e il teorema di Cramér-Wold — e l'ingegneria delle reti neurali per il controllo robotico — con le sue esigenze di efficienza computazionale, robustezza agli iperparametri e adattabilità a diversi ambienti.

Questa intersezione non è casuale: riflette una filosofia di ricerca che privilegia soluzioni con fondamenti formali solidi rispetto a soluzioni empiricamente efficaci ma teoricamente opache. Il fatto che questa filosofia produca, in LeWorldModel, risultati empiricamente competitivi — e su alcune metriche superiori — rispetto agli approcci puramente empirici, costituisce una validazione della tesi che la fondatezza teorica e l'efficacia pratica non siano obiettivi in conflitto ma, opportunamente combinati, si rinforzino a vicenda.

8.3 Limiti strutturali e direzioni future

Nonostante i risultati promettenti, LeWorldModel presenta limitazioni strutturali che definiscono chiaramente le direzioni di ricerca future più urgenti.

L'orizzonte di pianificazione limitato — conseguenza dell'accumulo di errori autoregressivi nel predittore — è probabilmente il limite più stringente per le applicazioni reali. La modellazione gerarchica del mondo, menzionata dagli

autori come direzione futura promettente, affronta questo problema decomponendo la pianificazione in livelli di astrazione: un livello alto che pianifica su orizzonti lunghi nello spazio degli obiettivi intermedi, e un livello basso che realizza ciascun obiettivo intermedio su orizzonti brevi nello spazio delle azioni primitive. Questa decomposizione richiederebbe la progettazione di una gerarchia di encoder e predittori con scale temporali diverse, un problema di architettura non banale che l'attuale LeWorldModel non affronta.

La dipendenza da dataset offline con sufficiente copertura delle dinamiche — un'assunzione che può essere difficile da soddisfare in ambienti reali dove alcune transizioni sono rare o costose da esplorare — costituisce un secondo limite strutturale. L'approccio reward-free non offre, per costruzione, un meccanismo per guidare l'esplorazione verso regioni sottorappresentate del dataset, il che limita la sua applicabilità a contesti dove si dispone di dati offline abbondanti e diversificati. Tecniche di pre-training su dataset video naturali su larga scala — come suggerito dagli autori per future estensioni — potrebbero fornire prior rappresentazionali che riducono la dipendenza da dati specifici del dominio.

La specifica dell'obiettivo attraverso immagini — necessaria per il costo latente $\|\hat{z}_H - z_g\|^2_2$ — è adeguata per i compiti goal-conditioned classici ma non generalizzabile a tutte le forme di specifica del compito. L'integrazione con modelli linguistici — per la specifica verbale degli obiettivi — o con strutture di ricompensa appresa — per la specifica attraverso feedback scalare — richiederebbe di connettere lo spazio latente di LeWorldModel con lo spazio semantico del linguaggio o con funzioni di ricompensa apprese, un problema di allineamento rappresentazionale di notevole complessità.

La performance sub-ottimale in ambienti con topologia non connessa — come Two-Room — indica che il prior gaussiano isotropico di SIGReg non è universalmente appropriato. La ricerca di distribuzioni target più flessibili, adattive alla struttura topologica dell'ambiente — attraverso tecniche come i modelli di flusso normalizzante (*normalizing flows*) o i prior di mixture — potrebbe estendere l'applicabilità del metodo senza sacrificare le garanzie formali, a patto di trovare formulazioni che mantengano la trattabilità computazionale e la semplicità degli iperparametri.

La limitazione relativa agli ambienti 3D complessi — dove DINO-WM mantiene un vantaggio per via delle rappresentazioni ricche provenienti dal pre-training su larga scala — suggerisce che il regime di addestramento da zero

su piccoli dataset è uno svantaggio strutturale rispetto agli approcci che sfruttano encoder pre-addestrati in ambienti visivamente ricchi. Una direzione promettente è l'inizializzazione dell'encoder di LeWorldModel con pesi pre-addestrati — ad esempio di DINOv2 — seguita da fine-tuning end-to-end con l'obiettivo LeWM. Questa approccio combinerebbe i vantaggi delle rappresentazioni pre-addestrate ricche con quelli dell'adattamento end-to-end al dominio specifico, potenzialmente superando entrambe le alternative pure.

8.4 Riflessioni conclusive sulla filosofia del metodo

L'aspetto forse più significativo di LeWorldModel, al di là dei risultati tecnici specifici, è la sua testimonianza di un approccio metodologico che privilegia la semplicità formale e la fondatezza teorica come guide per le scelte di progettazione. In un campo che ha prodotto un numero crescente di sistemi che funzionano empiricamente ma sono difficilmente comprensibili — architetture con decine di termini di perdita, iperparametri calibrati per tentativi su enormi cluster computazionali, procedure di stabilizzazione la cui efficacia dipende da sottigliezze implementative non documentate — LeWorldModel porta una prospettiva diversa: la complessità dell'obiettivo e la moltiplicazione degli iperparametri non sono garanzie di efficacia, ma sintomi di comprensione insufficiente del problema. Un regolarizzatore ben scelto, derivato da una teoria consolidata, può essere più efficace di una dozzina di penalità euristiche.

Questo principio — che la parsimonia formale e l'efficacia empirica non sono in conflitto, ma tendono ad allinearsi quando la formalizzazione è genuinamente adeguata al problema — è una delle lezioni più durature che la storia dell'apprendimento automatico offre, e che LeWorldModel ripropone in un contesto specifico con risultati convincenti. La strada verso sistemi di controllo autonomo capaci di pianificazione robusta e generalizzabile — la visione che anima l'intera tradizione dei modelli del mondo — passa, forse più di quanto si ritenga comunemente, attraverso la costruzione di rappresentazioni formalmente fondate piuttosto che attraverso l'accumulo di capacità empiricamente verificate ma teoricamente opache.

Supplemento al Capitolo 3 — Fondamenti Formali del Vision Transformer: Attenzione

Multi-Testa, Struttura degli Strati e Proprietà Computazionali

S3.1 Motivazione e collocazione del supplemento

La trattazione del Vision Transformer contenuta nel corpo principale della monografia si concentra sulle scelte architetturali di LeWorldModel — la configurazione tiny dell'encoder, la struttura del predittore, il condizionamento AdaLN — senza derivare il meccanismo di attenzione dal quale dipende tutta la capacità espressiva del sistema. Questa scelta è comprensibile in un testo orientato alla modellazione del mondo, ma lascia senza fondamento formale una componente che non è un dettaglio implementativo bensì il principio unificante dell'intera architettura. Il presente supplemento colma questa lacuna, derivando il meccanismo di attenzione multi-testa a partire dalle sue motivazioni informazionali, analizzandone le proprietà computazionali e di complessità, e raccordando la derivazione con le scelte specifiche di LeWorldModel.

S3.2 La rappresentazione dell'immagine come sequenza di token

Il primo passo trasformativo compiuto dall'encoder di LeWorldModel è la decomposizione dell'immagine in una sequenza di vettori discreti — i token — su cui il transformer può operare. Questa operazione, introdotta da Dosovitskiy e collaboratori nel 2021 con il lavoro originale sui Vision Transformer, risolve il problema fondamentale di applicare l'architettura transformer — concepita per sequenze monodimensionali di token linguistici — a oggetti bidimensionali con struttura spaziale.

Sia $o_t \in \mathbb{R}^{H\times W\times C}$ l'osservazione al tempo t, con $H = W = 224$ e $C = 3$. Si fissa una dimensione di patch $p = 14$ pixel. L'immagine viene partizionata in $N = (H/p)\cdot(W/p) = (224/14)^2 = 256$ patch non sovrapposte, ciascuna di dimensione $p\times p\times C = 14\times 14\times 3 = 588$ scalari. L'insieme delle patch è denotato $\{P_i\}_{i=1}^{N}$, dove $P_i \in \mathbb{R}^{p^2C}$ è il vettore ottenuto per appiattimento (flattening) della patch i-esima.

La proiezione lineare di ciascuna patch nella dimensione nascosta D del transformer è realizzata da una matrice di embedding $E \in \mathbb{R}^{D\times(p^2C)}$. L'i-esimo token di patch è:

$$x_i = E \cdot P_i + e_{pos}^i, \quad i = 1, \ldots, N$$

dove $e_{pos}^i \in \mathbb{R}^D$ è l'embedding posizionale appreso, specifico per la posizione i. Gli embedding posizionali sono parametri liberi ottimizzati congiuntamente con il resto del modello durante l'addestramento; contrariamente alla scelta sinusoidale originale dei transformer linguistici, il ViT utilizza embedding posizionali appresi, che risultano empiricamente più adatti al dominio visivo grazie alla loro adattabilità alla struttura delle correlazioni spaziali nel dataset specifico.

Il token classificatore [CLS] — un vettore appresi $e_{cls} \in \mathbb{R}^D$ — viene preposto alla sequenza di token di patch, producendo la sequenza di input al transformer:

$$Z^{(0)} = [e_{cls}, x_1, x_2, \ldots, x_N] \in \mathbb{R}^{(N+1)\times D}$$

dove la notazione $[\cdot, \cdot]$ indica la concatenazione lungo la dimensione delle posizioni. Per $N = 256$ e l'aggiunta del token classificatore, la sequenza ha lunghezza $N+1 = 257$.

La matrice di embedding di patch E contiene $D \times (p^2C) = 192 \times 588 = 112.896$ parametri. Gli embedding posizionali aggiungono $(N+1) \times D = 257 \times 192 = 49.344$ parametri. L'embedding del token classificatore aggiunge $D = 192$ parametri. La componente di embedding introduce quindi circa 162.432 parametri nell'encoder, ovvero circa il 3% dei parametri totali del modello.

S3.3 Il meccanismo di attenzione scalata: derivazione dalla prima principles

Il meccanismo di attenzione è il cuore del transformer. La sua derivazione può essere impostata a partire da un'esigenza informazionale precisa: dato un insieme di vettori (i token), ciascun token deve poter aggregare selettivamente informazione dagli altri token, con pesi di aggregazione che dipendono dal contenuto degli stessi token piuttosto che solo dalla loro posizione.

Formalmente, si vuole definire una mappa

$$\text{Attn}: \mathbb{R}^{n \times D} \to \mathbb{R}^{n \times D}$$

tale che l'output della posizione i sia una combinazione lineare degli input di tutte le posizioni, con coefficienti determinati dal contenuto:

$$\text{Attn}(Z)_i = \sum_{j=1}^{n} \alpha_{ij} \cdot (Z_j W_V)$$

dove i pesi $\alpha_{ij} \geq 0$ con $\sum_j \alpha_{ij} = 1$ codificano la rilevanza della posizione j per il calcolo dell'output alla posizione i, e $W_V \in \mathbb{R}^{D \times d_v}$ è una proiezione lineare che trasforma il token j nello spazio dei "valori". La scelta dei pesi α_{ij} deve rispondere al principio che posizioni con contenuto simile a i ricevano pesi alti e posizioni con contenuto dissimile ricevano pesi bassi.

Per ottenere questa proprietà si introduce il meccanismo query-key: si proietta Z in uno spazio delle "query" $Q = Z W_Q$ e in uno spazio delle "chiavi" $K = Z W_K$, con $W_Q, W_K \in \mathbb{R}^{D \times d_k}$. La similarità tra la query della posizione i e la chiave della posizione j è misurata dal prodotto scalare $q_i^\top k_j$, dove $q_i = (Z W_Q)_i \in \mathbb{R}^{d_k}$ è la query della posizione i e $k_j = (Z W_K)_j \in \mathbb{R}^{d_k}$ è la chiave della posizione j. I pesi di attenzione sono ottenuti normalizzando questi prodotti scalari con la funzione softmax:

$$\alpha_{ij} = \exp(q_i^\top k_j / \sqrt{d_k}) / \sum_l \exp(q_i^\top k_l / \sqrt{d_k})$$

Il fattore di scala $1/\sqrt{d_k}$ — introdotto da Vaswani e collaboratori nel 2017 nell'articolo fondante "Attention is All You Need" — previene la saturazione della softmax per valori grandi di d_k. Per comprenderlo formalmente, si supponga che i componenti di q e k siano indipendenti con media zero e varianza unitaria: il prodotto scalare $q^\top k = \sum_{l=1}^{d_k} q_l k_l$ è la somma di d_k variabili con media zero e varianza unitaria, e ha quindi varianza d_k. La sua deviazione standard è $\sqrt{d_k}$, e dividere per essa riporta il prodotto scalare a varianza unitaria, prevenendo che l'argomento della softmax cresca linearmente con d_k e rendendo i gradienti numericamente stabili.

In forma matriciale compatta, il meccanismo di attenzione scalata è:

$$\text{Attn}(Q, K, V) = \text{softmax}(QK^\top / \sqrt{d_k}) \cdot V$$

dove $Q = Z W_Q \in \mathbb{R}^{n \times d_k}$, $K = Z W_K \in \mathbb{R}^{n \times d_k}$, $V = Z W_V \in \mathbb{R}^{n \times d_v}$, e la softmax è applicata riga per riga alla matrice di score $QK^\top / \sqrt{d_k} \in \mathbb{R}^{n \times n}$.

S3.4 L'attenzione multi-testa: decomposizione e ricombinazione

Il meccanismo di attenzione a testa singola opera una sola proiezione delle query, delle chiavi e dei valori, il che limita la sua capacità di catturare simultaneamente relazioni di diversa natura tra i token. Per ovviare a questo, il ViT utilizza l'attenzione multi-testa (Multi-Head Self-Attention, MHSA): si eseguono h istanze parallele del meccanismo di attenzione, ciascuna con le proprie matrici di proiezione, e i loro output vengono concatenati e riproiettati.

Sia h il numero di teste e sia d_k = d_v = D/h la dimensione di ciascuna testa (nel caso standard della stessa dimensione per chiavi, query e valori). Per la testa m = 1, …, h:

Q^(m) = Z W_Q^(m), K^(m) = Z W_K^(m), V^(m) = Z W_V^(m) Head_m = softmax(Q^(m) K^(m)^⊤ / √d_k) · V^(m)

con W_Q^(m), W_K^(m), W_V^(m) ∈ ℝ^{D×d_k}. Gli output delle h teste vengono concatenati lungo la dimensione d_v:

MHSA(Z) = [Head_1, Head_2, …, Head_h] · W_O

dove [·, …, ·] denota la concatenazione che produce un tensore di dimensione n × (h · d_v) = n × D, e W_O ∈ ℝ^{D×D} è la matrice di output che proietta la concatenazione nella dimensione originale D.

La struttura multi-testa ha un'interpretazione precisa: ciascuna testa impara a prestare attenzione a sottospazi diversi dello spazio delle rappresentazioni. Una testa potrebbe specializzarsi nel catturare relazioni di prossimità spaziale tra patch adiacenti, un'altra nel cogliere dipendenze a lungo raggio tra patch distanti, e un'altra ancora nel raccogliere informazioni globali su regioni di colore simile. La ricombinazione tramite W_O consente al modello di integrare queste diverse tipologie di informazione relazionale in un'unica rappresentazione.

Per il ViT-T dell'encoder di LeWorldModel: D = 192, h = 3, d_k = D/h = 64. Le matrici di proiezione per ciascuna delle 3 teste hanno dimensione 192×64. Il costo dei parametri di MHSA per strato è:

W_Q, W_K, W_V: h · D · d_k = 3 · 3 · 192 · 64 = 3 · 36.864 = 110.592 parametri W_O: D · D = 192 · 192 = 36.864 parametri Bias: 3 · h · d_k + D = 3 · 3 · 64 + 192 = 768 parametri

Totale per l'MHSA di un strato: circa 148.224 parametri.

Per il predittore ViT-S con D_pred = 384 e h = 16: d_k = 384/16 = 24. Questa configurazione — insolita rispetto allo standard ViT-S che usa 6 teste con d_k = 64 — aumenta il numero di teste riducendo la dimensione per testa. La sua conseguenza è che il predittore può mantenere simultaneamente 16 forme diverse di attenzione — un vantaggio per la modellazione di dinamiche complesse — al prezzo di una dimensione per testa molto piccola, che riduce la capacità espressiva di ciascuna singola testa. La scelta riflette probabilmente un bilanciamento empirico specifico per il compito di previsione autoregressiva in spazio latente, dove la diversità delle teste (catturare relazioni di diverso tipo nella storia latente) è più preziosa della profondità di ciascuna testa.

S3.5 La struttura degli strati e la normalizzazione

Il transformer del ViT adotta la cosiddetta Pre-LayerNorm (Pre-LN): la normalizzazione è applicata prima di ogni blocco funzionale (attenzione o rete feed-forward), contrariamente alla Post-LN della formulazione originale di Vaswani, dove la normalizzazione segue il blocco.

La Layer Normalization (LayerNorm), introdotta da Ba, Kiros e Hinton nel 2016, normalizza ciascun vettore separatamente rispetto alla media e alla varianza dei propri elementi:

LayerNorm(x) = (x - μ_x) / √(σ²_x + ε) · γ + β

dove μ_x = (1/D) Σ_i x_i è la media degli elementi del vettore x ∈ ℝ^D, σ²_x = (1/D) Σ_i (x_i - μ_x)² è la varianza, ε > 0 è una costante di stabilità numerica (tipicamente 10^{-6}), e γ, β ∈ ℝ^D sono parametri appresi di scalatura e traslatura affine.

La LayerNorm differisce dalla Batch Normalization (BN) in un aspetto cruciale per il contesto di LeWorldModel: la prima normalizza ciascun vettore separatamente, indipendentemente dagli altri elementi del batch, mentre la seconda normalizza ciascuna componente attraverso gli elementi del batch. Questa distinzione non è meramente tecnica. La BN nella rete di proiezione

dell'encoder agisce sulla dimensione del batch, garantendo che la distribuzione degli embedding nel mini-batch sia centrata e a varianza controllata — una pre-condizionante che facilita l'azione di SIGReg, il quale proietta gli embedding nel batch su direzioni casuali e valuta la normalità delle proiezioni. Se la BN non fosse presente, la distribuzione degli embedding nel batch potrebbe avere media e varianza arbitrarie, complicando l'ottimizzazione di SIGReg. La LayerNorm nei blocchi del transformer serve invece uno scopo diverso: prevenire l'esplosione del gradiente e stabilizzare la dinamica dell'addestramento controllando la norma delle attivazioni all'interno di ciascun layer.

La struttura completa di uno strato Pre-LN del transformer è:

$$Z' = Z + \mathrm{MHSA}(\mathrm{LayerNorm}(Z)) \quad Z'' = Z' + \mathrm{FFN}(\mathrm{LayerNorm}(Z'))$$

dove la FFN (Feed-Forward Network) è una rete a due strati con funzione di attivazione GELU:

$$\mathrm{FFN}(x) = W_2 \cdot \mathrm{GELU}(W_1 x + b_1) + b_2$$

con $W_1 \in \mathbb{R}^{D_{ff} \times D}$, $b_1 \in \mathbb{R}^{D_{ff}}$, $W_2 \in \mathbb{R}^{D \times D_{ff}}$, $b_2 \in \mathbb{R}^D$, dove $D_{ff} = 4D$ è la dimensione del livello nascosto (la convenzione del fattore 4 è ereditata dalla letteratura sui transformer linguistici). Per il ViT-T: $D = 192$, $D_{ff} = 768$. I parametri della FFN per strato sono:

W_1, b_1: $D_{ff} \cdot D + D_{ff} = 4 \cdot 192^2 + 4 \cdot 192 = 147.456 + 768 = 148.224$
W_2, b_2: $D \cdot D_{ff} + D = 192 \cdot 768 + 192 = 147.456 + 192 = 147.648$

Totale FFN per strato: circa 295.872 parametri.

La funzione GELU (*Gaussian Error Linear Unit*), proposta da Hendrycks e Gimpel nel 2016, è definita come:

$$\mathrm{GELU}(x) = x \cdot \Phi(x)$$

dove $\Phi(x) = (1/2)[1 + \mathrm{erf}(x/\sqrt{2})]$ è la funzione di distribuzione cumulativa della gaussiana standard. L'intuizione è che GELU applica una scalatura al valore di input proporzionale alla probabilità che un'unità gaussiana standard sia inferiore a quel valore — equivalentemente, i valori molto negativi vengono soppressi quasi interamente, i valori molto positivi passano quasi invariati, e i valori intermedi sono smorzati con continuità. Rispetto a ReLU, GELU è

differenziabile ovunque e ha un comportamento più regolare nella regione di transizione, il che migliora la dinamica del gradiente nei transformer profondi.

S3.6 Conteggio dei parametri del ViT-T e del predittore ViT-S

Il conteggio preciso dei parametri consente di verificare la coerenza delle affermazioni architetturali della monografia e di comprendere l'effettiva distribuzione della capacità del modello tra i suoi componenti.

Per l'encoder ViT-T (D = 192, L = 12, h = 3):

Embedding di patch: D × (p^2C) = 192 × 588 = 112.896 Embedding posizionale: (N+1) × D = 257 × 192 = 49.344 Token CLS: D = 192

Per ciascuno dei L = 12 strati: LayerNorm_1 (2 × D): 2 × 192 = 384 MHSA (W_Q, W_K, W_V, W_O, bias): ~148.224 (vedi S3.4) LayerNorm_2 (2 × D): 384 FFN (W_1, b_1, W_2, b_2): ~295.872 Totale per strato: ~444.864

LayerNorm finale: 2 × D = 384

Totale encoder ViT-T: 162.432 + 12 × 444.864 + 384 ≈ 162.432 + 5.338.368 + 384 ≈ 5.501.184 ≈ 5,5M

Aggiungendo la rete di proiezione Proj_encoder con due strati lineari di dimensione 192×192 e 192×192, più la BatchNorm: W_in, b_in: 192 × 192 + 192 = 36.864 + 192 = 37.056 BN: 2 × 192 = 384 W_out, b_out: 192 × 192 + 192 = 37.056

Totale encoder (ViT-T + proiezione): circa 5.576.304 ≈ 5,6M parametri, coerente con il dato di "5 milioni" riportato nella monografia (la discrepanza è nella precisione del conteggio, non nell'ordine di grandezza).

Per il predittore ViT-S modificato (D_pred = 384, L_pred = 6, h = 16):

Per ciascuno dei 6 strati: MHSA (d_k = 24): h · (3 · D_pred · d_k + D_{pred}^2) = 16 · (3 · 384 · 24 + 0) + 384^2 ≈ 442.368 + 147.456 = 589.824 (Nota: W_O ∈ ℝ^{D_pred × D_pred} quindi D_{pred}^2 = 147.456 indipendentemente da h) FFN (D_ff = 4 × 384 = 1536): 2 × 384 × 1536 + 384 + 1536 ≈ 1.179.648 LayerNorm e bias: ~2.000 AdaLN MLP per ciascun strato: circa 384 × 2 × 384 × 2 = piccola aggiunta

Totale per strato: circa 1.771.472 Totale 6 strati: circa 10.628.832 ≈ 10,6M parametri

La stima è coerente con i "circa 10 milioni di parametri" riportati nella monografia.

Il totale del modello (encoder + predittore + proiezioni) è quindi circa 5,6M + 10,6M = 16,2M parametri, compatibile con la cifra di "circa 15 milioni" menzionata nel testo (la discrepanza riflette semplificazioni nel conteggio dei parametri delle reti di proiezione e dei bias).

S3.7 Complessità computazionale dell'attenzione e giustificazione della scelta architetturale

La complessità computazionale del meccanismo di attenzione è un fattore determinante nella progettazione del modello, in particolare per la fase di pianificazione dove il predittore viene valutato migliaia di volte per ciclo CEM.

Per un singolo strato di attenzione con sequenza di lunghezza n e dimensione D:

Calcolo di Q, K, V: $3 \cdot n \cdot D \cdot d_k \cdot h = 3 \cdot n \cdot D^2$ operazioni (con $d_k = D/h$) Prodotto $QK^\top$: $n \cdot n \cdot d_k \cdot h = n^2 \cdot D$ operazioni (uno per testa, scalato per h) Softmax: $n^2 \cdot h$ operazioni Prodotto con V: $n^2 \cdot D$ operazioni W_O: $n \cdot D^2$ operazioni FFN: $2 \cdot n \cdot D \cdot D_{ff} = 8n \cdot D^2$ operazioni

Costo dominante: $O(n^2 \cdot D + n \cdot D^2)$ per strato

Quando $n \ll D$ (sequenze corte), il termine $n \cdot D^2$ domina; quando $n \gg D$ (sequenze lunghe), il termine $n^2 \cdot D$ domina — la quadraticità nella lunghezza della sequenza è il classico collo di bottiglia del transformer standard.

Per il predittore di LeWorldModel, la sequenza in input ha lunghezza $N_{seq} = N \cdot d = 3 \cdot 1 = 3$ embedding latenti (con storia $N = 3$) — estremente corta. La dimensione nascosta è $D_{pred} = 384$. Pertanto $n = 3 \ll D_{pred} = 384$, e il costo è dominato da $n \cdot D_{pred}^2 = 3 \cdot 384^2 \approx 443K$ operazioni per strato, per 6 strati circa 2,7M operazioni per rollout. Moltiplicando per $H = 5$ passi di rollout: circa 13,5M operazioni per traiettoria candidata. Con $300 \times 30 = 9.000$ valutazioni per ciclo CEM su Push-T: circa $1{,}2 \times 10^{11}$ operazioni in virgola mobile. Su una GPU moderna con throughput di $\sim 10^{13}$ FLOP/s, questo

corrisponde a circa 0,01 secondi — un'indicazione di ordine di grandezza coerente con i tempi di pianificazione di circa 1 secondo osservati, dove la maggior parte del tempo è impiegata in overhead di scheduling, allocazione di memoria e trasferimento dati GPU.

Il confronto con DINO-WM è illuminante sulla questione computazionale. In quel sistema, la sequenza in input al predittore ha lunghezza N_seq = 256 token visivi (la griglia di patch, non il singolo token CLS). Con n = 256 e D = 768, il costo per strato è $O(n^2 \cdot D + n \cdot D^2) = O(256^2 \cdot 768 + 256 \cdot 768^2) = O(50M + 151M) \approx 200M$ operazioni. Per H = 5 passi e il numero equivalente di strati: circa 1B operazioni per traiettoria candidata, ovvero circa 75× più delle 13,5M di LeWorldModel. Questo differenziale — amplificato dagli overhead GPU su sequenze molto più lunghe — produce il fattore osservato sperimentalmente di circa 48-50 nella velocità di pianificazione.

S3.8 Il condizionamento AdaLN: derivazione e ruolo nell'architettura del predittore

Il condizionamento dell'azione nel predittore di LeWorldModel è realizzato attraverso l'Adaptive Layer Normalization (AdaLN), tecnica introdotta da Peebles e Xie nel 2023 nell'architettura DiT (*Diffusion Transformer*) per il condizionamento di transformer al rumore di diffusione e alla classe dell'immagine.

La motivazione è la seguente: si vuole che il predittore possa modulare le proprie rappresentazioni interne in funzione dell'azione a_t, ma senza introdurre il condizionamento come input aggiuntivo alla sequenza (che modificherebbe la lunghezza della sequenza e la struttura dell'attenzione) né come bias additivo uniforme (che non consentirebbe modulazioni specifiche per strato e per dimensione). AdaLN risolve questo problema generando i parametri affini della LayerNorm — scalatura γ e traslazione β — come funzioni dell'azione tramite una piccola rete neurale.

Formalmente, l'embedding dell'azione $a_t \in \mathbb{R}^A$ viene prima trasformato in un vettore di embedding $e_a = \text{MLP_action}(a_t) \in \mathbb{R}^D$ (un piccolo MLP con una singola layer nascosta), e poi per ciascuno degli L strati del predittore un MLP specifico dello strato genera i parametri affini:

$[\gamma_l, \beta_l] = \text{MLP}_l(e_a) \in \mathbb{R}^{2D}, l = 1, \ldots, L$

L'AdaLN per lo strato l è:

$$\text{AdaLN}_l(h, a) = \gamma_l(a) \odot \text{LayerNorm}(h) + \beta_l(a)$$

dove $\odot$ è il prodotto elemento per elemento e h è l'embedding dello strato.

L'inizializzazione zero — $\gamma_l(a) = 0$ e $\beta_l(a) = 0$ per ogni a all'inizio dell'addestramento — è cruciale e merita una spiegazione formale. Con questa inizializzazione, $\text{AdaLN}_l(h, a) = 0$ per ogni h e a all'inizio: il contributo condizionato è inizialmente nullo. La struttura residuale dello strato diventa:

$$Z' = Z + \text{MHSA}(\text{AdaLN}_1(Z, a)) = Z + \text{MHSA}(0) = Z + 0 = Z$$

Il predittore si comporta, nelle primissime iterazioni, come se non ci fosse condizionamento sull'azione — essenzialmente un transformer che processa solo la storia latente. Questo consente che le strutture della dinamica latente — le correlazioni temporali tra embedding consecutivi — vengano apprese nella fase iniziale dell'addestramento senza interferenze del condizionamento, il quale viene poi introdotto gradualmente man mano che i parametri MLP_l si aggiornano durante la retropropagazione.

Il contrasto con l'alternativa di concatenare l'embedding dell'azione ai token della sequenza è significativo: la concatenazione modificherebbe la lunghezza della sequenza (da N a N+1 token), alterando la struttura della maschera di attenzione causale, e introdurrebbe il condizionamento in modo uniforme attraverso tutti gli strati senza la modulazione strato-specifica che AdaLN consente. In ambienti dove l'azione influenza in modo diverso le diverse scale di astrazione della rappresentazione latente — ad esempio, alcune componenti latenti rispondono principalmente a azioni di alta frequenza (piccoli movimenti) e altre a azioni di bassa frequenza (cambi di direzione) — AdaLN offre la flessibilità di differenziare questo condizionamento tra gli strati.

S3.9 Il problema dell'exposure bias e la discrepanza addestramento-inferenza

Il regime di addestramento con teacher-forcing — in cui i target z_{t+1} sono sempre calcolati dall'encoder applicato all'osservazione reale, indipendentemente dalla qualità della previsione precedente — introduce una discrepanza strutturale tra la modalità di addestramento e quella di inferenza (pianificazione).

Durante l'addestramento, il predittore riceve in input embedding calcolati dall'encoder applicato a osservazioni reali:

input: $(z_t^{real}, a_t) = (enc_\theta(o_t), a_t)$ target: $z_{t+1}^{real} = enc_\theta(o_{t+1})$

Durante la pianificazione, il predittore opera in modalità autoregressiva: l'embedding in input al passo t+1 è l'embedding predetto al passo t, non calcolato da un'osservazione reale:

$\hat{z}_{t+1} = pred_\varphi(\hat{z}_t, a_t)$, $\hat{z}_1 = z_1^{real} = enc_\theta(o_1)$

La discrepanza è la seguente: il predittore non ha mai, durante l'addestramento, ricevuto in input embedding prodotti da un predittore imperfetto — ha sempre ricevuto embedding "ground-truth". In inferenza, gli errori di predizione si accumulano: $\hat{z}_{t+1}$ non è un embedding reale, ma la previsione di un embedding reale; il predittore non è stato addestrato a gestire questo tipo di errori nella propria distribuzione di input.

Questo fenomeno — noto nella letteratura come *exposure bias* e analizzato da Bengio e collaboratori nel 2015 nel contesto dei modelli sequence-to-sequence — ha una conseguenza formale sul comportamento del rollout autoregressivo. Si possono definire la distribuzione degli embedding reali:

$P_Z = \{P_{z_t^{real}}\}_{t=1}^T$

e la distribuzione degli embedding predetti in inferenza:

$P_{\hat{Z}} = \{P_{\hat{z}_t}\}_{t=1}^T$

La discrepanza $P_Z \neq P_{\hat{Z}}$ è garantita in presenza di qualsiasi errore di predizione non nullo, e tende ad amplificarsi con l'orizzonte: $\hat{z}_{t+k}$ è la k-esima applicazione del predittore a partire da un embedding contaminato da errore, e l'errore si propaga e amplifica lungo la catena. Il degrado della qualità della previsione con l'orizzonte osservato sperimentalmente nella monografia — particolarmente marcato in OGBench-Cube — è una conseguenza diretta di questo accumulo.

Una soluzione parziale all'exposure bias, esplorata nella letteratura dei modelli linguistici, è il *scheduled sampling*: durante l'addestramento, con probabilità crescente nel corso delle epoche, si sostituisce il target ground-truth con la previsione del passo precedente. Questo espone progressivamente il predittore

alla propria distribuzione di errori, riducendo la discrepanza con la modalità di inferenza. Il costo è la maggiore instabilità dell'addestramento nelle fasi iniziali, dove le previsioni sono di qualità insufficiente da usare come input. LeWorldModel non adotta questa tecnica, privilegiando la stabilità del teacher-forcing, e accetta l'accumulo di errore come compromesso.

Un'analisi formale dell'accumulo di errore può essere condotta nel caso lineare approssimato. Sia pred_φ(z, a) ≈ A z + B a la linearizzazione locale del predittore, e sia η_t = ẑ_t - z_t^real l'errore di predizione al passo t. Allora:

η_{t+1} = ẑ_{t+1} - z_{t+1}^real = pred_φ(ẑ_t, a_t) - pred_φ(z_t^real, a_t) + (pred_φ(z_t^real, a_t) - z_{t+1}^real) ≈ A η_t + ε_t

dove ε_t = pred_φ(z_t^real, a_t) - z_{t+1}^real è l'errore del modello applicato all'embedding reale (la perdita di predizione al passo t), supposto piccolo ma non nullo. La norma dell'errore soddisfa:

‖η_{t+k}‖ ≤ ‖A‖^k ‖η_t‖ + (‖A‖^{k-1} + ‖A‖^{k-2} + … + 1) · sup_t ‖ε_t‖

Se ‖A‖ < 1 (predittore contrattivo), l'errore rimane limitato; se ‖A‖ ≥ 1 (predittore espansivo), l'errore può crescere esponenzialmente con l'orizzonte k. Nella pratica, il predittore appreso non è garantito essere contrattivo in tutte le direzioni dello spazio latente, il che spiega la degradazione osservata per orizzonti lunghi e motiva la scelta di un orizzonte H = 5 relativamente breve nel ciclo MPC.

S3.10 La maschera di attenzione causale nel predittore

Il predittore di LeWorldModel deve prevedere ẑ_{t+1} a partire dalla storia z_{t-N+1:t}, rispettando la proprietà fondamentale di causalità: la previsione al passo t deve dipendere solo dagli embedding nei passi precedenti a t, non dai passi successivi. Questa proprietà è implementata attraverso una maschera di attenzione triangolare inferiore.

La matrice di score S = Q K^⊤ / √d_k ∈ ℝ^{n×n} viene modificata impostando a -∞ tutte le entrate superiori alla diagonale:

S_{ij}^{masked} = S_{ij} se i ≥ j = -∞ se i < j

In questo modo, softmax(S^{masked})_{ij} = 0 per i < j: il token alla posizione i non può prestare attenzione ai token nelle posizioni j > i. Dopo l'applicazione della softmax, la matrice di attenzione risultante è triangolare inferiore:

A = softmax(S^{masked}), A_{ij} = 0 per i < j

Questa struttura garantisce che la previsione $\hat{z}_{t+1} = \text{pred}_\varphi(z_{t-N+1:t}, a_t)$ sia calcolata dalla posizione t+1 della sequenza di output, la quale può solo prestare attenzione alle posizioni 1, …, t+1 nella sequenza di input — ovvero agli embedding z_{t-N+1}, z_{t-N+2}, …, z_t e all'embedding dell'azione a_t, non a embedding temporalmente successivi.

La struttura causale è fondamentale per due ragioni. Da un lato, garantisce che il predittore non possa "spiare" il futuro durante l'addestramento, il che porterebbe a un overfitting ai pattern specifici del dataset piuttosto che all'apprendimento di una struttura predittiva genuina. Dall'altro, rende il modello compatibile con la modalità di inferenza autoregressiva: la stessa architettura usata durante l'addestramento con teacher-forcing può essere applicata iterativamente durante la pianificazione, senza modifiche strutturali.

S3.11 Connessione con il teorema della rappresentazione universale

Una domanda naturale è se l'architettura ViT-T con 12 strati di attenzione sia sufficientemente espressiva da apprendere encoder arbitrariamente complessi della distribuzione delle osservazioni. Il teorema di approssimazione universale per i transformer — dimostrato in forme diverse da Yun e collaboratori nel 2020 e da Pérez e collaboratori nel 2021 — fornisce una risposta affermativa:

per ogni funzione misurabile $f: \mathcal{X}^n \to \mathcal{X}^n$ e per ogni $\varepsilon > 0$, esiste un transformer con larghezza e profondità sufficienti tale che la funzione implementata disti da f meno di ε in norma L^2.

La dimostrazione si basa su due componenti: la capacità della softmax attention di implementare una funzione di look-up approssimata (il softmax, per basse temperature, si approssima alla mappa argmax, che può selezionare elementi specifici della sequenza di input), e la capacità della FFN di approssimare funzioni continue arbitrarie (per il teorema di approssimazione universale delle reti neurali). Combinando questi due elementi, un transformer

sufficientemente ampio e profondo può implementare qualsiasi algoritmo di trasformazione della sequenza esprimibile come composizione di look-up e trasformazioni continue.

Tuttavia, questa garanzia di universalità è di natura esistenziale — garantisce l'esistenza di parametri con buona approssimazione, non che la procedura di addestramento li trovi. Per il ViT-T con 5,5M di parametri e la distribuzione dei dati di LeWorldModel, la capacità è probabilmente adeguata per i compiti considerati — come suggerisce l'evidenza empirica — ma la garanzia formale di convergenza all'ottimizzatore globale non è disponibile per modelli neurali di questa complessità.

Appendice A — Teoria delle Funzioni Caratteristiche, Statistica di Epps-Pulley e Fondamenti Matematici della Regolarizzazione SIGReg

A.1 Premessa: la logica della derivazione

La regolarizzazione SIGReg è presentata nel corpo della monografia come un regolarizzatore che impone la gaussianità degli embedding proiettati, con un riferimento al teorema di Cramér-Wold come fondamento della garanzia anti-collasso e alla statistica di Epps-Pulley come strumento di misura della deviazione dalla gaussiana standard. Questa presentazione, pur corretta nelle sue affermazioni principali, lascia senza svolgimento le derivazioni matematiche che ne costituiscono la giustificazione più profonda: perché la funzione caratteristica caratterizza univocamente una distribuzione; perché la proiezione su direzioni casuali è sufficiente per il matching di distribuzioni multidimensionali; come la statistica di Epps-Pulley si calcola concretamente e qual è la sua forma esplicita; qual è il legame con la Maximum Mean Discrepancy. Questa appendice svolge sistematicamente tutte queste derivazioni.

A.2 Funzioni caratteristiche: definizione e proprietà fondamentali

Sia X una variabile aleatoria a valori reali con distribuzione di probabilità P_X. La *funzione caratteristica* di X è definita come il valore atteso della funzione e^{itX} per t ∈ ℝ:

$$\varphi_X(t) = \mathbb{E}[e^{itX}] = \int_{-\infty}^{+\infty} e^{itx}\, dP_X(x)$$

dove i è l'unità immaginaria. Questa definizione è ben posta per ogni distribuzione di probabilità, senza alcun requisito sull'esistenza di momenti o sulla struttura della distribuzione — la funzione caratteristica esiste sempre, poiché $|e^{itx}| = 1$ per ogni x, t reali. In questo si distingue dalla funzione generatrice dei momenti $M_X(t) = \mathbb{E}[e^{tX}]$, che può non esistere per distribuzioni con code pesanti.

Le proprietà fondamentali della funzione caratteristica, che si dimostreranno utili nel seguito, sono:

Valori iniziali: $\varphi_X(0) = \mathbb{E}[e^{i\cdot 0\cdot X}] = \mathbb{E}[1] = 1$. La funzione caratteristica vale sempre 1 nell'origine.

Continuità uniforme: φ_X è uniformemente continua su ℝ. Questo segue dalla disuguaglianza $|\varphi_X(t+h) - \varphi_X(t)| \le \mathbb{E}[|e^{i(t+h)X} - e^{itX}|] = \mathbb{E}[|e^{ihX} - 1|]$, che tende a 0 per $h \to 0$ per il teorema della convergenza dominata (poiché $|e^{ihX} - 1| \le 2 \in L^1$).

Inversione (teorema di Lévy): Se $\varphi_X \in L^1(\mathbb{R})$, allora P_X ha densità f_X con: $f_X(x) = (1/2\pi) \int_{-\infty}^{+\infty} e^{-itx}\, \varphi_X(t)\, dt$

ovvero la funzione caratteristica è la trasformata di Fourier della densità e la densità è l'antitrasformata.

Unicità (teorema di Lévy-Cramér): La corrispondenza tra distribuzioni di probabilità e funzioni caratteristiche è biunivoca:

$$P_X = P_Y \Leftrightarrow \varphi_X(t) = \varphi_Y(t) \text{ per ogni } t \in \mathbb{R}$$

Questo teorema — il più importante nella teoria delle funzioni caratteristiche — afferma che la funzione caratteristica *determina univocamente* la distribuzione. La dimostrazione si basa sulla formula di inversione di Lévy: se $\varphi_X = \varphi_Y$, allora le due distribuzioni hanno la stessa trasformata di Fourier, e per l'unicità della trasformata di Fourier devono coincidere come distribuzioni.

Convergenza debole: Una successione di distribuzioni (P_n) converge debolmente a P se e solo se le funzioni caratteristiche convergono puntualmente: $\varphi_{X_n}(t) \to \varphi_X(t)$ per ogni $t \in \mathbb{R}$. Questo è il teorema di continuità di Lévy, che trasforma la convergenza debole — difficile da verificare direttamente — in convergenza puntuale delle funzioni caratteristiche, molto più trattabile.

Indipendenza: Se X e Y sono indipendenti, $\varphi_{X+Y}(t) = \varphi_X(t) \cdot \varphi_Y(t)$. Questo segue immediatamente dalla fattorizzazione del valore atteso per variabili indipendenti.

A.3 La funzione caratteristica della distribuzione gaussiana

Il calcolo della funzione caratteristica della distribuzione gaussiana standard $N(0,1)$ è un esempio paradigmatico di come le proprietà analitiche della gaussiana consentano calcoli esatti in contesti dove altre distribuzioni richiedono approssimazioni numeriche.

Per $X \sim N(0,1)$ con densità $f(x) = (1/\sqrt{(2\pi)})\, e^{-x^2/2}$:

$$\varphi_X(t) = \mathbb{E}[e^{itX}] = (1/\sqrt{(2\pi)}) \int_{-\infty}^{+\infty} e^{itx}\, e^{-x^2/2}\, dx$$

Si completa il quadrato nell'esponente:

$$itx - x^2/2 = -(x^2 - 2itx)/2 = -((x - it)^2 + t^2)/2 = -(x-it)^2/2 - t^2/2$$

Quindi: $\varphi_X(t) = (1/\sqrt{(2\pi)})\, e^{-t^2/2} \int_{-\infty}^{+\infty} e^{-(x-it)^2/2}\, dx$

L'integrale restante è $\int_{-\infty}^{+\infty} e^{-(x-it)^2/2}\, dx = \sqrt{(2\pi)}$ per il prolungamento analitico dell'integrale gaussiano nel piano complesso (il teorema di Cauchy garantisce che l'integrale lungo la retta reale traslata di it nel piano complesso coincida con l'integrale gaussiano standard). Pertanto:

$$\varphi_X(t) = e^{-t^2/2}$$

Per $X \sim N(\mu, \sigma^2)$: poiché $X = \mu + \sigma Z$ con $Z \sim N(0,1)$, si ha $\varphi_X(t) = e^{i\mu t}\, \varphi_Z(\sigma t) = e^{i\mu t - \sigma^2 t^2/2}$.

Per un vettore aleatorio $X \sim N(0, I_d)$ in $\mathbb{R}^d$, le componenti sono indipendenti e ciascuna è $N(0,1)$. La funzione caratteristica multivariata è:

$\varphi_X(t) = \mathbb{E}[e^{it^\top X}] = \Pi_{j=1}^d \varphi_{X_j}(t_j) = \Pi_{j=1}^d e^{-t_j^2/2} = e^{-\|t\|^2/2}$

Questa identità — $\varphi_{N(0,I_d)}(t) = e^{-\|t\|^2/2}$ — è la caratterizzazione analitica della gaussiana isotropica in $\mathbb{R}^d$.

A.4 Il teorema di Cramér-Wold: enunciato e dimostrazione

Il teorema di Cramér-Wold è il pilastro matematico su cui si fonda la strategia di matching proiettato di SIGReg. La sua affermazione precisa è la seguente:

Teorema di Cramér-Wold: Siano P e Q due distribuzioni di probabilità su $\mathbb{R}^d$. Allora:

$P = Q \Leftrightarrow \forall\ u \in \mathbb{R}^d: u^\top X \sim u^\top Y$

dove $X \sim P$ e $Y \sim Q$. In altre parole, due distribuzioni multidimensionali coincidono se e solo se tutte le loro proiezioni unidimensionali coincidono.

Dimostrazione: Il verso ($\Rightarrow$) è triviale: se $P = Q$ allora per ogni funzione misurabile g, $\mathbb{E}_P[g(X)] = \mathbb{E}_Q[g(X)]$, e applicando $g = e^{i\alpha \cdot}$ per ogni α si ottiene l'identità delle proiezioni.

Per il verso ($\Leftarrow$), si sfrutta la caratterizzazione delle distribuzioni attraverso le funzioni caratteristiche. Supponiamo che $u^\top X \sim u^\top Y$ per ogni $u \in \mathbb{R}^d$. Allora per ogni $u \in \mathbb{R}^d$ e ogni $s \in \mathbb{R}$:

$\mathbb{E}[e^{is(u^\top X)}] = \mathbb{E}[e^{is(u^\top Y)}]$

ovvero $\varphi_X(su) = \varphi_Y(su)$ per ogni $u \in \mathbb{R}^d$ e $s \in \mathbb{R}$. Poiché per ogni $t \in \mathbb{R}^d$ si può scrivere $t = su$ con $u = t/\|t\|$ e $s = \|t\|$ (per $t \neq 0$), si ottiene:

$\varphi_X(t) = \varphi_Y(t)$ per ogni $t \in \mathbb{R}^d$ (e ovviamente per $t = 0$ entrambe valgono 1)

Per il teorema di unicità (Lévy-Cramér multivariato), $\varphi_X = \varphi_Y$ implica $P = Q$. □

Risultato di convergenza debole di SIGReg (Maes et al., 2026, Appendice A). Sia $Z \in \mathbb{R}^{N\times B\times d}$ il tensore degli embedding latenti,

u^(m) ∈ S^{d−1} direzioni casuali unitarme, e h^(m) := Zu^(m) le proiezioni scalari. La statistica di Epps-Pulley univariata è

$$T^{(m)} = \int w(t)\, |\varphi_N(t; h^{(m)}) - \varphi_0(t)|^2 \, dt ,$$

dove φ_N(t; h) = (1/N) Σ_n e^{ith_n} è la funzione caratteristica empirica, φ_0(t) = e^{−t²/2} è la funzione caratteristica della $\mathcal{N}(0,1)$, e w(t) = e^{−t²/(2λ²)} è una funzione di ponderazione gaussiana. Il regolarizzatore SIGReg è la media su M direzioni casuali: SIGReg(Z) = (1/M) Σ_{m=1}^{M} T^(m). Per il teorema di Cramér-Wold, nel limite asintotico M → ∞ vale il seguente risultato di convergenza debole:

$$\text{SIGReg}(Z) \to 0 \Leftrightarrow P_Z \to \mathcal{N}(0, I) ,$$

che fornisce la garanzia formale di non-collasso: qualsiasi forma di collasso (compreso il collasso parziale su un sottospazio) produce SIGReg(Z) > 0, indipendentemente dalla struttura specifica della degenerazione. Nell'implementazione pratica, l'integrale è calcolato numericamente con una regola di quadratura (ad es. trapezi con nodi uniformi in [0.2, 4]).

Corollario per SIGReg: Una distribuzione P_Z su $\mathbb{R}^d$ coincide con N(0, I_d) se e solo se per ogni u ∈ S^{d-1}, la variabile u^⊤ Z segue una distribuzione N(0, 1). Questo è il fondamento dell'approccio: per verificare che P_Z = N(0, I_d), è sufficiente verificare la normalità univariata di ogni proiezione.

Osservazione sulla sufficienza delle proiezioni unitarie: Il teorema di Cramér-Wold richiede le proiezioni per ogni u ∈ $\mathbb{R}^d$, non solo per ‖u‖ = 1. Tuttavia, se u^⊤ Z ~ N(0,1) per ogni u ∈ S^{d-1}, allora per ogni α > 0:

$$(\alpha u)^\top Z = \alpha(u^\top Z) \sim N(0, \alpha^2)$$

che coincide con (αu)^⊤ G per G ~ N(0, I_d), poiché (αu)^⊤ G ~ N(0, α² ‖u‖²) = N(0, α²). Quindi le proiezioni unitarie sono sufficienti.

A.5 La funzione caratteristica empirica

Per un campione {x_1, x_2, …, x_n} estratto da una distribuzione sconosciuta P_X, la *funzione caratteristica empirica* (ECF) è:

$\hat{\varphi}_n(t) = (1/n)\ \Sigma_{k=1}^n e^{itx_k} = (1/n)\ \Sigma_{k=1}^n \cos(tx_k) + i \cdot (1/n)\ \Sigma_{k=1}^n \sin(tx_k)$

La parte reale è $\mathrm{Re}[\hat{\varphi}_n(t)] = (1/n)\ \Sigma_k \cos(tx_k)$ e la parte immaginaria è $\mathrm{Im}[\hat{\varphi}_n(t)] = (1/n)\ \Sigma_k \sin(tx_k)$.

La ECF è uno stimatore della vera funzione caratteristica $\varphi_X(t) = \mathbb{E}[e^{itX}]$. Per la legge dei grandi numeri, $\hat{\varphi}_n(t) \to \varphi_X(t)$ quasi certamente per ogni $t \in \mathbb{R}$. La velocità di convergenza è data dal teorema del limite centrale applicato alle variabili e^{itX_k}: per ogni t fissato, $\sqrt{n}\ (\hat{\varphi}_n(t) - \varphi_X(t))$ converge in distribuzione a una variabile complessa gaussiana con varianza $\mathrm{Var}[e^{itX}] = \varphi_X(2t) - |\varphi_X(t)|^2$.

Un vantaggio tecnico fondamentale della ECF come base per i test statistici di distribuzione è che il modulo della ECF è bounded: $|\hat{\varphi}_n(t)| \leq 1$ per ogni t e ogni realizzazione del campione, il che consente stime uniformemente stabili indipendentemente dalla distribuzione generatrice.

Il modulo quadrato della ECF:

$|\hat{\varphi}_n(t)|^2 = \mathrm{Re}[\hat{\varphi}_n(t)]^2 + \mathrm{Im}[\hat{\varphi}_n(t)]^2 = (1/n^2)\ \Sigma_{j=1}^n \Sigma_{k=1}^n e^{it(x_j - x_k)}$

Poiché i termini con $j = k$ danno $e^{0} = 1$ e i termini con $j \neq k$ sono coppie coniugate, si ha:

$|\hat{\varphi}_n(t)|^2 = (1/n^2)\ [n + \Sigma_{j\neq k}\ e^{it(x_j - x_k)}] = (1/n) + (1/n^2)\ \Sigma_{j\neq k}\ e^{it(x_j - x_k)}$

Prendendo la parte reale, e notando che la parte immaginaria di $\Sigma_{j\neq k}\ e^{it(x_j - x_k)}$ è zero per antisimmetria:

$|\hat{\varphi}_n(t)|^2 = (1/n) + (1/n^2)\ \Sigma_{j\neq k}\ \cos(t(x_j - x_k)) = (1/n^2)\ \Sigma_{j,k}\ \cos(t(x_j - x_k))$

dove l'ultima uguaglianza include i termini diagonali ($j=k$), per i quali $\cos(0) = 1$.

A.6 La statistica di Epps-Pulley: derivazione formale

Thomas W. Epps e Lawrence B. Pulley proposero nel 1983, sulla rivista *Biometrika*, una statistica di test di normalità basata sulla distanza integrata pesata tra la funzione caratteristica empirica e quella della distribuzione gaussiana standard.

Data la funzione caratteristica target della N(0,1), $\varphi_0(t) = e^{-t^2/2}$, e la ECF $\hat{\varphi}_n(t)$ del campione, la statistica di Epps-Pulley è:

$$T_n^{EP} = n \int_{-\infty}^{+\infty} |\hat{\varphi}_n(t) - \varphi_0(t)|^2 \, w(t) \, dt$$

dove w(t) è una funzione peso integrabile non negativa. Il fattore n è una normalizzazione che rende la distribuzione asintotica di T_n^{EP} sotto l'ipotesi nulla ($P_X = N(0,1)$) non degenere.

Si sviluppa il modulo quadrato integrato:

$$|\hat{\varphi}_n(t) - \varphi_0(t)|^2 = |\hat{\varphi}_n(t)|^2 - 2\,\mathrm{Re}[\hat{\varphi}_n(t)]\,\varphi_0(t) + |\varphi_0(t)|^2 = |\hat{\varphi}_n(t)|^2 - 2\,\mathrm{Re}[\hat{\varphi}_n(t)]\,e^{-t^2/2} + e^{-t^2}$$

Integrando con il peso $w(t) = e^{-t^2/2}$ (una scelta classica che attribuisce peso maggiore alle frequenze basse, più influenzate dalla forma centrale della distribuzione, e peso minore alle frequenze alte, più influenzate dalle code):

$$\int |\hat{\varphi}_n(t) - \varphi_0(t)|^2 \, e^{-t^2/2} \, dt = \int |\hat{\varphi}_n(t)|^2 \, e^{-t^2/2} \, dt - 2 \int \mathrm{Re}[\hat{\varphi}_n(t)] \, e^{-t^2} \, dt + \int e^{-t^2} \, e^{-t^2/2} \, dt$$

Ciascun termine può essere calcolato esplicitamente sfruttando le formule integrali per funzioni esponenziali e trigonometriche.

Primo termine: Sostituendo $|\hat{\varphi}_n(t)|^2 = (1/n^2) \sum_{j,k} \cos(t(x_j - x_k))$:

$$\int_{-\infty}^{+\infty} |\hat{\varphi}_n(t)|^2 \, e^{-t^2/2} \, dt = (1/n^2) \sum_{j,k} \int_{-\infty}^{+\infty} \cos(t(x_j - x_k)) \, e^{-t^2/2} \, dt$$

L'integrale $\int_{-\infty}^{+\infty} \cos(as) \, e^{-s^2/(2\sigma^2)} \, ds = \sigma\sqrt{(2\pi)} \, e^{-a^2\sigma^2/2}$ è una formula standard (si dimostra completando il quadrato nell'esponente, come fatto per la funzione caratteristica gaussiana). Con $\sigma = 1$ e $a = x_j - x_k$:

$$\int_{-\infty}^{+\infty} \cos(t(x_j - x_k)) \, e^{-t^2/2} \, dt = \sqrt{(2\pi)} \, e^{-(x_j - x_k)^2/2}$$

Quindi il primo termine è:

$(1/n^2)\ \Sigma_{j,k}\ \sqrt{(2\pi)}\ e^{-(x_j-x_k)^2/2} = \sqrt{(2\pi)}/n^2\ \Sigma_{j,k}\ e^{-(x_j-x_k)^2/2}$

Secondo termine: Sostituendo $\mathrm{Re}[\hat{\varphi}_n(t)] = (1/n)\ \Sigma_j \cos(tx_j)$:

$\int_{-\infty}^{+\infty} Re[\hat{\varphi}_n(t)]\ e^{-t^2}\ dt = (1/n)\ \Sigma_j \int_{-\infty}^{+\infty} \cos(tx_j)\ e^{-t^2}\ dt$

Con $\sigma^2 = 1/2$ e $a = x_j$: $\int_{-\infty}^{+\infty} \cos(tx_j)\ e^{-t^2}\ dt = \sqrt{\pi}\ e^{-x_j^2/4}$

Quindi il secondo termine è:

$-2 \cdot (1/n)\ \Sigma_j\ \sqrt{\pi}\ e^{-x_j^2/4} = -(2\sqrt{\pi}/n)\ \Sigma_j\ e^{-x_j^2/4}$

Terzo termine: $\int_{-\infty}^{+\infty} e^{-3t^2/2}\ dt = \sqrt{(2\pi/3)}$

Raccogliendo tutti i termini, la statistica di Epps-Pulley assume la forma chiusa:

$T_n^{EP} = n\ [\sqrt{(2\pi)}/n^2\ \Sigma_{j,k}\ e^{-(x_j-x_k)^2/2} - (2\sqrt{\pi}/n)\ \Sigma_j\ e^{-x_j^2/4} + \sqrt{(2\pi/3)}]$

$= \sqrt{(2\pi)}/n\ \Sigma_{j,k}\ e^{-(x_j-x_k)^2/2} - 2\sqrt{\pi}\ \Sigma_j\ e^{-x_j^2/4} + n\sqrt{(2\pi/3)}$

Questa forma chiusa è computazionalmente notevole: il calcolo richiede $O(n^2)$ operazioni per la doppia somma (il termine dominante), senza alcun integrale numerico e senza nodi di quadratura. Tuttavia, la forma chiusa è specifica per la scelta $w(t) = e^{-t^2/2}$. Per scelte diverse del peso, o per la variante numerica usata in LeWorldModel dove i nodi di quadratura permettono flessibilità nella forma del peso, si ricorre alla formula approssimata:

$T \approx \Sigma_{k=1}^{K} w_k\ |\hat{\varphi}_n(t_k) - e^{-t_k^2/2}|^2$

con nodi $\{t_k\}_{k=1}^{K}$ *e pesi* $\{w_k\}$ *della quadratura di Newton-Cotes o di Gauss. L'errore di approssimazione della quadratura è* $O(h^{p+1}\ \|T^{(p+1)}\|_{\infty}\ (b-a))$ per la regola di ordine p sull'intervallo [a, b] con passo $h = (b-a)/K$, e per la regola dei trapezi ($p = 2$) con $K = 10$ nodi nell'intervallo [0.2, 4] il passo è $h = 0.38$ e l'errore è dell'ordine di $O(h^3) = O(0.055)$, accettabile per l'uso come termine di perdita nell'ottimizzazione.

A.7 Il calcolo del gradiente di SIGReg per la retropropagazione

L'ottimizzazione di SIGReg via discesa del gradiente richiede il calcolo del gradiente ∂ SIGReg / ∂ Z rispetto alla matrice degli embedding $Z \in \mathbb{R}^{n\times d}$. Questo gradiente è necessario per propagare il segnale di addestramento dall'obiettivo SIGReg all'encoder tramite retropropagazione.

Nella versione con quadratura numerica, $\mathrm{SIGReg}(Z) = (1/M)\ \Sigma_{m=1}^{M} T^{(m)}$, dove $T^{(m)}$ è la statistica di Epps-Pulley calcolata sulla proiezione scalare $h^{(m)} = Z\, u^{(m)} \in \mathbb{R}^n$.

Per calcolare $\partial T^{(m)}/\partial Z$, si nota che $T^{(m)}$ dipende da Z solo attraverso $h^{(m)} = Z\, u^{(m)}$, con:

$$\partial T^{(m)}/\partial Z = (\partial T^{(m)}/\partial h^{(m)}) \cdot (\partial h^{(m)}/\partial Z)^{\top}$$

Il termine $\partial h^{(m)}/\partial Z_{ij} = u^{(m)}_j \cdot \delta_{ik}$ (dove δ è il delta di Kronecker), ovvero $\partial h^{(m)}_i/\partial Z_{ij} = u^{(m)}_j$. Quindi:

$$(\partial T^{(m)}/\partial Z)_{ij} = (\partial T^{(m)}/\partial h^{(m)}_i) \cdot u^{(m)}_j$$

e in forma matriciale:

$$\partial T^{(m)}/\partial Z = (\partial T^{(m)}/\partial h^{(m)}) \cdot (u^{(m)})^{\top} \in \mathbb{R}^{n\times d}$$

dove $\partial T^{(m)}/\partial h^{(m)} \in \mathbb{R}^n$ è il gradiente della statistica di Epps-Pulley rispetto al campione scalare $h^{(m)}$.

Per la statistica con quadratura:

$$\partial T^{(m)}/\partial h^{(m)}_i = 2\ \Sigma_{k=1}^{K} w_k\ \mathrm{Re}[(\hat{\varphi}_n(t_k) - e^{-t_k^2/2}) \cdot \partial\hat{\varphi}_n(t_k)/\partial h^{(m)}_i]$$

dove $\partial\hat{\varphi}_n(t_k)/\partial h^{(m)}_i = (i t_k/n)\, e^{i t_k h^{(m)}_i}$. Prendendo la parte reale:

$$\partial T^{(m)}/\partial h^{(m)}_i = (2/n)\ \Sigma_{k=1}^{K} w_k\ t_k\ [\mathrm{Im}[\hat{\varphi}_n(t_k)] \cos(t_k h^{(m)}_i) - \mathrm{Re}[\hat{\varphi}_n(t_k)] \sin(t_k h^{(m)}_i) + e^{-t_k^2/2} \sin(t_k h^{(m)}_i)]$$

Questa espressione è calcolabile in $O(nK)$ operazioni per direzione m, e per M direzioni in $O(MnK)$ operazioni, che con $M = 1024$, $n = 512$, $K = 10$ è circa 5×10^6 operazioni — computazionalmente trascurabile rispetto ai costi del passaggio avanti del transformer.

La moltiplicazione esterna $\partial T^{(m)}/\partial Z = \partial T^{(m)}/\partial h^{(m)} \cdot (u^{(m)})^\top$ richiede $n \times d = 512 \times 192 \approx 10^5$ operazioni per direzione, e per $M = 1024$ direzioni circa 10^8 operazioni — ancora inferiore al costo del passaggio avanti dell'encoder.

Il gradiente totale di SIGReg è:

$$\partial \text{SIGReg} / \partial Z = (1/M) \sum_{m=1}^{M} (\partial T^{(m)}/\partial h^{(m)}) \cdot (u^{(m)})^\top$$

che può essere calcolato in forma vettorizzata come:

$$\partial \text{SIGReg} / \partial Z = (1/M)\, G_h \cdot U^\top$$

dove $G_h \in \mathbb{R}^{n \times M}$ ha $G_h[:, m] = \partial T^{(m)}/\partial h^{(m)}$ e $U \in \mathbb{R}^{d \times M}$ ha $U[:, m] = u^{(m)}$. Il prodotto $G_h \cdot U^\top \in \mathbb{R}^{n \times d}$ può essere calcolato in $O(nMd)$ operazioni tramite moltiplicazione matriciale, sfruttando i BLAS level-3 ottimizzati sulle GPU.

A.8 Connessione con la Maximum Mean Discrepancy

La Maximum Mean Discrepancy (MMD) è una misura di distanza tra distribuzioni definita rispetto a uno spazio di funzioni campione (*witness function space*) $\mathcal{H}$:

$$\text{MMD}(P, Q; \mathcal{H}) = \sup_{f \in \mathcal{H},\, \|f\|_{\mathcal{H}} \leq 1} |\mathbb{E}_{X\sim P}[f(X)] - \mathbb{E}_{Y\sim Q}[f(Y)]|$$

Per spazi di Hilbert riproduttivi (RKHS) con kernel κ, si può mostrare che:

$$\text{MMD}^2(P, Q; \kappa) = \mathbb{E}_{X,X'\sim P}[\kappa(X,X')] - 2\mathbb{E}_{X\sim P, Y\sim Q}[\kappa(X,Y)] + \mathbb{E}_{Y,Y'\sim Q}[\kappa(Y,Y')]$$

La connessione con la statistica di Epps-Pulley proiettata emerge scegliendo come kernel κ il kernel gaussiano:

$$\kappa(x, y) = e^{-(x-y)^2/(2\sigma^2)}$$

Si può mostrare, attraverso l'espansione di κ in serie di Fourier e il confronto con la formula di integrazione della statistica di Epps-Pulley, che:

$$\int_{-\infty}^{+\infty} |\hat{\varphi}_n(t) - \varphi_0(t)|^2 e^{-\sigma^2 t^2/2}\, dt \propto \mathit{MMD}^2_n(\hat{P}_n, N(0,1); \kappa_\sigma)$$

dove $\hat{P}_n$ è la distribuzione empirica del campione e $\varkappa_{\sigma}(x,y) = e^{-(x-y)^2/(2\sigma^2)}$. La costante di proporzionalità dipende solo da σ e da costanti della distribuzione target, non dal campione.

Per dimostrare questa connessione, si espande la MMD^2 empirica con kernel gaussiano:

$$MMD^2_n(\hat{P}_n, N(0,1); \varkappa_{\sigma}) = (1/n^2)\ \Sigma_{j,k}\ e^{-(x_j-x_k)^2/(2\sigma^2)} - (2/n)\ \Sigma_j\ \mathbb{E}_{Y\sim N(0,1)}[e^{-(x_j-Y)^2/(2\sigma^2)}] + \mathbb{E}_{Y,Y'\sim N(0,1)}[e^{-(Y-Y')^2/(2\sigma^2)}]$$

Il secondo termine: $\mathbb{E}_{Y\sim N(0,1)}[e^{-(x-Y)^2/(2\sigma^2)}] = \int e^{-(x-y)^2/(2\sigma^2)}\ (1/\sqrt{(2\pi)})\ e^{-y^2/2}\ dy$. Completando il quadrato:

$$-(x-y)^2/(2\sigma^2) - y^2/2 = -[y^2(1+\sigma^2/\sigma^2\cdot 1/\sigma^2 + 1/2) - \dots]$$

Si esegue il calcolo sistematicamente. Sia $a = 1/(2\sigma^2)$ e $b = 1/2$ i coefficienti dell'esponente in y:

$$-(x-y)^2/(2\sigma^2) - y^2/2 = -x^2/(2\sigma^2) + xy/\sigma^2 - y^2/(2\sigma^2) - y^2/2 = -x^2/(2\sigma^2) + xy/\sigma^2 - y^2(1/(2\sigma^2) + 1/2) = -x^2/(2\sigma^2) + xy/\sigma^2 - y^2(1+\sigma^2)/(2\sigma^2)$$

Completando il quadrato in y:

$$= -x^2/(2\sigma^2) + (1/\sigma^2)^2\ x^2\ /\ (2(1+\sigma^2)/\sigma^2) - (1+\sigma^2)/(2\sigma^2)\ (y - x/(1+\sigma^2))^2$$

$$= -\ x^2/(2(\sigma^2+1)) - (1+\sigma^2)/(2\sigma^2)\ (y - x/(1+\sigma^2))^2$$

Integrando rispetto a y:

$$\int\ (1/\sqrt{(2\pi)})\ e^{-x^2/(2(\sigma^2+1))}\ e^{-(1+\sigma^2)/(2\sigma^2)(y-x/(1+\sigma^2))^2}\ dy = e^{-x^2/(2(\sigma^2+1))}\cdot(\sigma/\sqrt{(1+\sigma^2)})$$

Pertanto: $\mathbb{E}_{Y\sim N(0,1)}[e^{-(x-Y)^2/(2\sigma^2)}] = (\sigma/\sqrt{(1+\sigma^2)})\ e^{-x^2/(2(1+\sigma^2))}$

Il terzo termine: $\mathbb{E}_{Y,Y'\sim N(0,1)}[e^{-(Y-Y')^2/(2\sigma^2)}] = \iint\ (1/2\pi)\ e^{-(y-y')^2/(2\sigma^2)}\ e^{-y^2/2-y'^2/2}\ dy\ dy'$. Per simmetria e la formula sopra con $x = 0$: questo vale $\sigma/\sqrt{(1+\sigma^2)}\cdot(\sigma/\sqrt{(1+\sigma^2)}) = \sigma^2/(1+\sigma^2)$.

Raccogliendo:

$$MMD^2_n \propto (1/n^2)\ \Sigma_{j,k}\ e^{-(x_j-x_k)^2/(2\sigma^2)} - (2\sigma)/(n\sqrt{(1+\sigma^2)})\ \Sigma_j\ e^{-x_j^2/(2(1+\sigma^2))} + \sigma^2/(1+\sigma^2)$$

Per $\sigma = 1$ (kernel gaussiano standard): $MMD^2_n \propto (1/n^2) \Sigma_{j,k} e^{-(x_j-x_k)^2/2} - (\sqrt{2}/n) \Sigma_j e^{-x_j^2/4} + 1/2$.

Confrontando con la forma chiusa della statistica di Epps-Pulley (sez. A.6):

$T_n^{EP} = \sqrt{(2\pi)}/n \Sigma_{j,k} e^{-(x_j-x_k)^2/2} - 2\sqrt{\pi} \Sigma_j e^{-x_j^2/4} + n\sqrt{(2\pi/3)}$

Le due espressioni hanno la stessa struttura: una somma di kernel kernel gaussiani su tutte le coppie di punti del campione, meno due volte una somma di kernel tra il campione e il target, più una costante. Il fattore di proporzionalità e le costanti differiscono per via del peso e della normalizzazione, ma la struttura informazionale è identica. SIGReg, attraverso la statistica di Epps-Pulley proiettata, è quindi equivalente a una MMD proiettata: una stima Monte Carlo della MMD tra la distribuzione degli embedding e la gaussiana isotropica standard, calcolata in modo computazionalmente trattabile attraverso la riduzione dimensionale proiettiva.

A.9 Proprietà asintotiche e analisi statistica della garanzia anti-collasso

La garanzia anti-collasso di SIGReg è di natura asintotica: essa vale nel limite $M \to \infty$ del numero di proiezioni. Per M finito, SIGReg = 0 non implica necessariamente $P_Z = N(0, I_d)$, ma solo che le M proiezioni campionate non rivelano deviazioni dalla gaussianità. La probabilità che M proiezioni casuali non rivelino una deviazione effettivamente presente dipende dalla struttura della deviazione e da M.

Per distribuzioni che non siano gaussiane isotropiche, esiste almeno una direzione $u^* \in S^{d-1}$ lungo cui la proiezione $u^{*\top} Z$ non è gaussiana standard. La probabilità che nessuna delle M proiezioni campionate cada in un cono ε-vicino a u^* è $(1 - C_d \varepsilon^{d-1})^M$ per qualche costante C_d che dipende dalla geometria della sfera d-dimensionale. Per $M >> C_d^{-1} \varepsilon^{-(d-1)}$, questa probabilità è trascurabile, e SIGReg rivelerà la deviazione con alta probabilità.

Per d = 192 (la dimensione latente di LeWorldModel) e M = 1024, il rilevamento di deviazioni con struttura di codimensione bassa — lungo una singola direzione u^* — richiede $M >> C_{192}^{-1} \varepsilon^{-191}$, un requisito impossibile da soddisfare con M = 1024. Questo suggerisce che SIGReg con

M = 1024 non può rilevare con alta probabilità deviazioni che si manifestano solo lungo una singola direzione nello spazio 192-dimensionale.

Tuttavia, questa limitazione teorica è meno stringente di quanto appaia, per due ragioni. In primo luogo, il collasso rilevante in pratica non è una deviazione monodimensionale ma una concentrazione degli embedding su un sottospazio di dimensione bassa — una struttura che si manifesta in molte direzioni contemporaneamente, aumentando la probabilità di rilevamento. In secondo luogo, il gradiente di SIGReg fornisce, durante l'ottimizzazione, una pressione continua verso la gaussianità indipendentemente dalla struttura esatta della deviazione: anche se M = 1024 proiezioni non sono sufficienti per una garanzia statistica di alta confidenza, sono sufficienti per generare un gradiente orientato nella direzione di riduzione del collasso ad ogni iterazione.

L'insensibilità empirica di LeWorldModel al valore di M (il tasso di successo rimane sostanzialmente stabile per $M \in [64, 1024]$) è coerente con questa interpretazione: ciò che conta non è la precisione statistica del test di normalità, ma la presenza di un gradiente informativo che contrasti il collasso ad ogni passo dell'ottimizzazione.

A.10 Distribuzione asintotica sotto l'ipotesi nulla

La distribuzione asintotica della statistica di Epps-Pulley (scalata da n) sotto l'ipotesi nulla H_0: $P_X = N(0,1)$, ovvero la distribuzione che T_n^{EP} segue quando il campione è effettivamente gaussiano standard, è stata derivata da Epps e Pulley nel loro articolo originale del 1983.

Nel limite $n \to \infty$, la distribuzione di T_n^{EP} sotto H_0 converge a quella di una forma quadratica di un processo gaussiano:

$$T_n^{EP} \xrightarrow{d} \Sigma_{j=1}^{\infty} \lambda_j Z_j^2$$

dove $\{Z_j\}_{j=1}^{\infty}$ sono variabili gaussiane indipendenti standard e $\{\lambda_j\}$ sono gli autovalori di un operatore integrale associato al kernel:

$$K(s, t) = \int (e^{isx} - \varphi_0(s))(e^{itx} - \varphi_0(t))^* w(s)w(t)\, dP_X(x)$$

L'espressione degli autovalori può essere calcolata esplicitamente per la gaussiana target e la scelta di peso standard, e i valori tabulati forniscono i quantili del test per soglie di significatività prefissate. Per l'uso di SIGReg

nell'addestramento del modello, tuttavia, la distribuzione asintotica sotto l'ipotesi nulla non è rilevante: non si esegue un test statistico formale di ipotesi ma si usa la statistica come termine di perdita in un'ottimizzazione. L'unica proprietà che conta è che SIGReg → 0 quando P_Z → N(0, I_d), il che è garantito dalla consistenza della statistica di Epps-Pulley come test della normalità gaussiana.

A.11 Generalizzazioni e distribuzioni target alternative

La struttura matematica di SIGReg non è intrinsecamente legata alla gaussiana isotropica come distribuzione target. Qualsiasi distribuzione target P* per cui la funzione caratteristica sia nota in forma chiusa (o calcolabile numericamente) può essere sostituita nella definizione della statistica:

$$T^{(m)} = \Sigma_{k=1}^K w_k \, |\hat{\varphi}_n(t_k; h^{(m)}) - \varphi_{P_*^{proj}}(t_k)|^2$$

dove $P_^{proj}$ *è la distribuzione delle proiezioni di P* lungo le direzioni casuali $u^{(m)}$.

Per la gaussiana isotropica N(0, I_d), le proiezioni $u^\top Z$ lungo direzioni unitarie seguono N(0, 1) e $\varphi_{N(0,1)}(t) = e^{-t^2/2}$.

Per una miscela di gaussiane — un target più flessibile adatto a spazi degli stati con struttura multimodale come Two-Room — le proiezioni di una miscela K-component $\Sigma_k \pi_k N(\mu_k, \Sigma_k)$ lungo la direzione u sono una miscela univariata $\Sigma_k \pi_k N(u^\top \mu_k, u^\top \Sigma_k u)$, con funzione caratteristica:

$$\varphi_{mix}(t; u) = \Sigma_k \pi_k e^{it u^\top \mu_k - t^2 u^\top \Sigma_k u / 2}$$

Questa forma è ancora calcolabile in forma chiusa, consentendo una versione di SIGReg con target a miscela. Il costo è l'introduzione dei parametri della miscela (medie, covarianze, pesi) come iperparametri aggiuntivi o, alternativamente, come parametri appresi dai dati, il che aumenta la complessità del sistema.

Per distribuzioni su varietà compatte — come la sfera S^{d-1} o il toro $T^d = S^1 \times \dots \times S^1$ — esistono teorie analoghe delle funzioni caratteristiche (trasformate di Fourier su gruppi compatti) che potrebbero essere sfruttate per costruire varianti di SIGReg adatte a spazi latenti con struttura geometrica non

euclidea. Tuttavia, il calcolo di queste varianti richiederebbe lo svolgimento esplicito della decomposizione armonica sul gruppo specifico, un problema matematicamente non banale che va oltre l'approccio di SIGReg nella sua versione attuale.

Appendice B — Il Metodo dell'Entropia Incrociata: Derivazione Variazionale, Teoria della Convergenza e Confronto con le Alternative

B.1 Premessa concettuale: il problema dell'ottimizzazione stocastica in spazio di azione

L'impiego di metodi di ottimizzazione stocastica per la ricerca della sequenza di azioni ottimale nel paradigma MPC non è, come potrebbe sembrare, una concessione pragmatica alla difficoltà dell'ottimizzazione non convessa, bensì una scelta metodologicamente fondata che riflette le proprietà strutturali del problema. Il paesaggio della perdita latente $\mathscr{C}(\hat{z}_H) = \|\hat{z}_H - z_g\|^2_2$ come funzione della sequenza di azioni a_{1:H} è, in generale, altamente non convesso: il rollout autoregressivo del predittore introduce non-linearità a ogni passo, e la composizione di H funzioni non lineari produce una mappa $a_\{1:H\} \to \hat{z}_H$ la cui struttura è quasi sempre irriducibilmente complicata. In questo regime, i metodi di discesa del gradiente deterministici tendono a impuntarsi in minimi locali di qualità scadente, particolarmente quando l'ottimizzazione inizia da un punto casuale senza informazione a priori sulla struttura del paesaggio. I metodi stocastici, per contro, esplorano il paesaggio campionando sequenze di azioni da una distribuzione che viene aggiornata iterativamente, il che consente di sfuggire ai minimi locali e di convergere verso regioni di costo basso anche in assenza di informazione sul gradiente.

Il Cross-Entropy Method (CEM), proposto nella sua forma originale da Reuven Rubinstein negli anni '90 per la stima di probabilità di eventi rari e poi sviluppato come algoritmo di ottimizzazione da Rubinstein e Kroese nel 2004, appartiene alla famiglia dei metodi di ottimizzazione bayesiana basata su modello (*model-based optimization*) nella misura in cui mantiene esplicitamente un modello probabilistico della distribuzione delle soluzioni candidate e lo

aggiorna in funzione delle osservazioni. La sua derivazione dal principio di minima divergenza di Kullback-Leibler conferisce al metodo una giustificazione variazionale precisa che trascende la sua presentazione euristica come semplice stima dei quantili superiori.

B.2 Derivazione variazionale dal principio di minima KL-divergenza

Si consideri il problema di trovare il minimizzatore globale della funzione di costo $\mathscr{C}$: $\mathscr{A}^H \to \mathbb{R}$, dove $\mathscr{A}^H$ è lo spazio delle sequenze di azioni di orizzonte H. Si definisce la distribuzione di Gibbs (o distribuzione di Boltzmann) associata al costo $\mathscr{C}$ con temperatura $\tau > 0$:

$\pi_{\tau}(a_{1:H}) \propto \exp(-\mathscr{C}(a_{1:H})/\tau)$

La funzione di partizione è $Z_\tau = \int_{\mathscr{A}^H} \exp(-\mathscr{C}(a_{1:H})/\tau)\, d\, a_{1:H}$, e la normalizzazione garantisce che π_τ sia una distribuzione di probabilità. Nel limite $\tau \to 0$, π_τ converge alla distribuzione di Dirac $\delta(a_{1:H} - a^*)$, dove $a^* = \operatorname{argmin} \mathscr{C}(a_{1:H})$ è il minimizzatore globale — assumendo per semplicità che sia unico. Per $\tau \to \infty$, π_τ converge alla distribuzione uniforme su $\mathscr{A}^H$.

L'idea fondatrice del CEM è trovare una distribuzione parametrica q_θ, scelta in una famiglia trattabile $\mathscr{F} = \{q_\theta : \theta \in \Theta\}$, che sia il più vicina possibile a π_τ nel senso della divergenza KL. Il problema variazionale è:

$\theta^* = \operatorname{argmin}_{\theta \in \Theta} KL(\pi_\tau \,\|\, q_\theta)$

dove la divergenza KL è definita come:

$KL(\pi_\tau \,\|\, q_\theta) = \int \pi_\tau(a) \log \pi_\tau(a)\, da - \int \pi_\tau(a) \log q_\theta(a)\, da = H(\pi_\tau) + CE(\pi_\tau, q_\theta)$

La prima parte, $H(\pi_\tau) = -\int \pi_\tau(a) \log \pi_\tau(a)\, da$, è l'entropia di π_τ e non dipende da θ. La seconda parte, $CE(\pi_\tau, q_\theta) = -\int \pi_\tau(a) \log q_\theta(a)\, da$, è la cross-entropia tra π_τ e q_θ, ed è l'unica componente che dipende da θ. Pertanto:

$\theta^* = \operatorname{argmin}_{\theta \in \Theta} CE(\pi_\tau, q_\theta) = \operatorname{argmax}_{\theta \in \Theta} \mathbb{E}_{a \sim \pi_\tau}[\log q_\theta(a)]$

La derivazione dell'aggiornamento iterativo richiede ora di approssimare l'aspettazione $\mathbb{E}_{a \sim \pi_\tau}[\log q_\theta(a)]$ in modo computazionalmente trattabile, poiché il campionamento diretto da π_τ è in generale impossibile (richiede la conoscenza di Z_τ).

Si ricorre all'*importance sampling*: se q_0 è una distribuzione proposta da cui è facile campionare, si ha:

$$\mathbb{E}_{a \sim \pi_\tau}[\log q_\theta(a)] = \mathbb{E}_{a \sim q_0}[W(a) \log q_\theta(a)]$$

dove $W(a) = \pi_\tau(a)/q_0(a) \propto \exp(-\mathscr{C}(a)/\tau)/q_0(a)$ è il peso di importanza non normalizzato. Poiché la costante di normalizzazione Z_τ cancella nell'ottimizzazione rispetto a θ, si ha:

$$\theta^* = \operatorname{argmax}_{\theta \in \Theta} \mathbb{E}_{a \sim q_0}[\exp(-\mathscr{C}(a)/\tau) \log q_\theta(a)]$$

Questa è la forma operativa dell'aggiornamento del CEM: si campionano N sequenze $a^{(1)}, \ldots, a^{(N)}$ dalla distribuzione proposta corrente q_0, si calcolano i pesi di importanza $w_i \propto \exp(-\mathscr{C}(a^{(i)})/\tau)$, e si ottimizza q_θ per massimizzare la verosimiglianza pesata:

$$\theta^* = \operatorname{argmax}_{\theta \in \Theta} (1/N) \Sigma_{i=1}^{N} w_i \log q_\theta(a^{(i)})$$

Per la famiglia gaussiana multivariata $q_\theta = \mathcal{N}(\mu, \Sigma_a)$, la soluzione in forma chiusa è la media e la covarianza ponderata:

$$\mu^{new} = (\Sigma_i w_i a^{(i)}) / (\Sigma_i w_i) \quad \Sigma_a^{new} = (\Sigma_i w_i (a^{(i)} - \mu^{new})(a^{(i)} - \mu^{new})^\top) / (\Sigma_i w_i)$$

Questa è la soluzione della massima verosimiglianza pesata per una gaussiana — il passo fondamentale dell'algoritmo CEM.

B.3 L'approssimazione elitista come limite $\tau \to 0$

Nell'implementazione pratica del CEM, invece di usare pesi esponenziali $w_i = \exp(-\mathscr{C}(a^{(i)})/\tau)$, si adotta l'approssimazione *elitista*: si assegna peso uniforme alle K sequenze con costo più basso (*elite*) e peso zero alle rimanenti $N - K$:

$$w_i = 1 \text{ se } a^{(i)} \in \mathscr{E}, \ 0 \text{ altrimenti}$$

dove $\mathcal{E} = \{i : \mathcal{C}(a^{(i)}) \leq \mathcal{C}_{(K)}\}$ e $\mathcal{C}_{(K)}$ è il K-esimo ordine-statistica del costo tra le N sequenze campionate.

Si può mostrare che questa approssimazione corrisponde al limite $\tau \to 0$ con rapporto K/N fissato: per $\tau \to 0$, la distribuzione di Gibbs π_τ concentra la propria massa sui $K/N \cdot 100\%$ più bassi del costo, ovvero sulle sequenze che si trovano nel quantile inferiore $\varrho = K/N$. La scelta elitista con $\varrho = 0.1$ (10% delle sequenze) corrisponde all'approssimazione di una distribuzione fortemente concentrata attorno ai minimi.\

Gli aggiornamenti diventano quindi:

$\mu^{new} = (1/K) \Sigma_{i \in \mathcal{E}} a^{(i)}$ $\Sigma_a^{new} = (1/(K-1)) \Sigma_{i \in \mathcal{E}} (a^{(i)} - \mu^{new})(a^{(i)} - \mu^{new})^\top$

Nella configurazione di LeWorldModel: N = 300, K = 30, $T_{iter} = 30$ (Push-T) o 10 (altri ambienti), con struttura diagonale per Σ_a (si stimano solo le varianze marginali, non le covarianze) per motivi di stabilità statistica con K = 30 campioni.

B.4 Analisi della convergenza

La convergenza del CEM verso il minimizzatore globale di $\mathcal{C}$ non è garantita in generale. Le analisi disponibili nella letteratura riguardano casi specifici o forniscono garanzie di convergenza più deboli del tipo "convergenza alla distribuzione concentrata sull'ottimo" piuttosto che "convergenza al parametro ottimale".

Il risultato fondamentale è il seguente. Si assuma che $\mathcal{C} : \mathcal{A}^H \to \mathbb{R}$ sia misurabile e che esista un minimizzatore a^* con valore ottimale $C^* = \mathcal{C}(a^*)$. Si assuma che la famiglia $\mathcal{M}(\mu, \Sigma_a)$ abbia supporto su $\mathcal{A}^H$ per ogni $\mu \in \mathcal{A}^H$ e ogni Σ_a definita positiva. Allora, sotto opportune condizioni di regolarità, per ogni $\varepsilon > 0$:

$\mathbb{P}(|\mu_t - a^*| \leq \varepsilon) \to 1$ per $t \to \infty$

dove μ_t è la media della distribuzione al passo t, assumendo che sia il numero di campioni N che il numero di iterazioni T_{iter} crescano opportunamente.

La prova di questo risultato si articola in due parti. La prima garantisce che, per N sufficientemente grande, la stima degli argomenti elite converge alla

distribuzione vera nel quantile ϱ (law of large numbers per l'empirical CDF). La seconda mostra che, quando la distribuzione è sufficientemente concentrata attorno al minimizzatore, l'aggiornamento gaussiano mantiene la concentrazione nell'intorno del minimo.

Più precisamente, si definisce il raggio di concentrazione al passo t come:

$$r_t = \mathbb{E}_{a \sim q_t}[\|a - a^*\|^2]^{1/2}$$

e si vuole mostrare che $r_t \to 0$. Sotto la gaussiana isotropica con varianza σ_t^2, $r_t \approx \sigma_t \sqrt{H \cdot d_a}$. La varianza viene aggiornata come:

$$\sigma_{t+1}^2 = (1/(K-1)) \Sigma_{i \in \mathcal{E}_t} (a^{(i)} - \mu_t^{new})^2$$

Assumendo che $\mu_t \approx a^*$ (che si verifica asintoticamente) e che le elite siano le K campioni da $\mathcal{N}(a^*, \sigma_t^2 I)$ con $\mathcal{C}$-valore più basso, la varianza σ_{t+1}^2 è la varianza del quantile inferiore ϱ di una gaussiana con varianza σ_t^2. Per la distribuzione del minimo ordine-statistica di una gaussiana con varianza σ^2, la varianza delle K-elite decresce con σ_t^2 per un fattore $c_\varrho < 1$ che dipende solo da $\varrho = K/N$:

$$\sigma_{t+1}^2 \approx c_\varrho \cdot \sigma_t^2$$

con $c_\varrho = \mathrm{Var}_{X \sim \mathcal{N}(0,1)}[X \mid X \le Q_{1-\varrho}(\mathcal{N}(0,1))]$, che per $\varrho = 0.1$ è circa 0.58. Questo produce la convergenza geometrica $r_t \approx r_0 \cdot c_\varrho^{t/2} \to 0$.

In pratica, la convergenza del CEM dipende crucialmente dall'inizializzazione (la distribuzione proposta iniziale deve avere supporto sufficiente nella regione degli ottimi) e dalla struttura del paesaggio del costo. Per paesaggi con molti minimi locali comparabili, il CEM può convergere a soluzioni sub-ottime nella cui regione di attrazione cade la distribuzione iniziale.

B.5 Struttura computazionale del CEM nel contesto MPC

Nel ciclo MPC di LeWorldModel, un'iterazione dell'algoritmo CEM procede come segue. Si inizializza la distribuzione $q_0 = \mathcal{N}(0, \sigma_{init}^2 I_{H \cdot d_a})$, con $\sigma_{init}^2 = 1$ e d_a la dimensione dell'azione. Si campionano $N = 300$ sequenze di azioni $A = \{a^{(i)}_{1:H}\}_{i=1}^N$ dalla distribuzione corrente q_t.

Per ogni sequenza a^(i)_{1:H}, si esegue il rollout del predittore per H = 5 passi a partire dall'embedding corrente z_1:

$\hat{z}_1^{(i)} = z_1$, $\hat{z}_{s+1}^{(i)} = \text{pred}_\varphi(\hat{z}_s^{(i)}, a_s^{(i)})$ per s = 1, ..., H

e si calcola il costo terminale $\mathscr{C}^{(i)} = \|\hat{z}_H^{(i)} - z_g\|_2^2$. Si identificano le K = 30 elite, e si aggiornano i parametri della gaussiana secondo le formule della sezione B.2. Si ripete per T_{iter} iterazioni, e il piano eseguito è μ_{T_{iter}}.

Il costo computazionale totale per ciclo CEM è:

Costo_totale = N · T_{iter} · [Costo_rollout(H) + Costo_costo]

Il costo del rollout per H passi è H moltiplicazioni matrice-vettore nel predittore ViT-S, ciascuna O($D_{pred}^2 \cdot n_{seq}$) con D_{pred} = 384 e n_{seq} = 3 token: circa $H \cdot 3 \cdot 384^2 \approx 1.7$M operazioni. Con N · T_{iter} = 300 · 30 = 9.000 rollout: circa 1.5 × 10^10 operazioni, eseguibili su GPU in ≈ 0.8−1.0 secondi grazie alla parallelizzazione sulle N sequenze. L'aggiornamento dei parametri gaussiani richiede O(N · H · d_a) = O(300 · 5 · 2) = 3.000 operazioni per iterazione — trascurabile.

B.6 Model Predictive Path Integral: derivazione e confronto con CEM

Il Model Predictive Path Integral (MPPI), introdotto da Williams, Aldrich e Theodorou nel 2015 e poi sviluppato in diverse varianti, è un algoritmo di ottimizzazione stocastica concettualmente imparentato con il CEM ma derivato da una prospettiva diversa: quella del controllo ottimale stocastico con massimizzazione dell'entropia.

La derivazione di MPPI parte dalla formulazione path-integral del controllo ottimale. Si considera una traiettoria stocastica $\hat{z}_{1:H}$ nello spazio latente generata dalla politica perturbata:

$a_t = \bar{u}_t + \varepsilon_t$, $\varepsilon_t \sim \mathcal{N}(0, \Sigma_\varepsilon)$

dove $\bar{u}_{1:H}$ è la sequenza di controllo nominale e ε_t è un rumore di esplorazione gaussiano. L'obiettivo è trovare $\bar{u}_{1:H}$ che minimizzi il costo medio:

$$J(\bar{u}) = \mathbb{E}_{\{\varepsilon_{1:H}\}}[\mathscr{C}(\hat{z}_H) + \Sigma_{t=1}^{H-1} (1/2)\, \bar{u}_t^{\top} \Sigma_{\varepsilon}^{-1} \bar{u}_t]$$

dove il secondo termine è un costo quadratico sul controllo nominale che regolarizza la soluzione. Questo problema può essere scritto nella forma di un'aspettazione rispetto alla distribuzione delle traiettorie perturbate:

$$J(\bar{u}) = -\lambda \log \mathbb{E}_{\{\varepsilon \sim \mathcal{N}(0, \Sigma_\varepsilon)\}}[\exp(-\mathscr{C}(\hat{z}_H)/\lambda)]$$

dove $\lambda = \Sigma_\varepsilon \cdot \|\bar{u}\|^2$ è la temperatura di importanza. La soluzione ottimale in forma path-integral è:

$$\bar{u}\{t\} = \bar{u}_t + \mathbb{E}\{\tau \sim \mathbb{Q}^\}[\varepsilon_t]$$

dove $\mathbb{Q}^*$ è la distribuzione di traiettorie con peso $\exp(-\mathscr{C}(\hat{z}_H)/\lambda)$. Questa aspettazione può essere approssimata con importance sampling:

$$\bar{u}^*\{1{:}H\} = \Sigma\{i=1\}^N \omega_i\, (\bar{u}_{1:H} + \varepsilon^{(i)}_{1:H}) \,/\, \Sigma_j \omega_j$$

dove $\omega_i = \exp(-(1/\lambda)(\mathscr{C}^{(i)} - C_{min}))$ sono i pesi normalizzati per stabilità numerica con $C_{min} = \min_i \mathscr{C}^{(i)}$.

Il confronto tra MPPI e CEM rivela differenze strutturali rilevanti. MPPI aggiorna la media della distribuzione con una media pesata *di tutte* le N sequenze, pesate con i pesi esponenziali; CEM aggiorna solo con le K elite, scartando le N − K rimanenti. In termini di utilizzo dell'informazione, MPPI è più efficiente: usa il contributo di ogni sequenza campionata, anche quelle con costo alto, attribuendo loro peso basso ma non nullo. CEM scarta completamente il 90% delle sequenze, un'operazione che introduce un bias nella stima della distribuzione ottimale.

Dall'analisi variazionale, MPPI corrisponde al problema di minimizzazione:

$$\theta^* = \operatorname{argmin}_{\theta} KL(\mathbb{Q}_\lambda \,\|\, q_\theta)$$

dove $\mathbb{Q}_\lambda$ è la distribuzione di Gibbs con temperatura λ, e la minimizzazione è rispetto a q_θ nella direzione *inversa* della KL (cioè KL(target ‖ model)) rispetto a quella usata nel CEM standard (KL(model ‖ target)). Questa differenza nella direzione della KL ha conseguenze pratiche: KL(p ‖ q) penalizza le regioni dove q è piccolo ma p è grande (*zero-avoiding*), mentre KL(q ‖ p) penalizza le regioni dove p è grande ma q è piccolo (*zero-forcing*). L'approssimazione MPPI tende a coprire tutti i modi della distribuzione ottimale, mentre CEM tende a concentrarsi sul modo più alto.

In ambienti con un singolo minimo globale dominante — come quelli considerati in LeWorldModel — questa differenza è irrilevante e le due famiglie di metodi producono soluzioni di qualità comparabile. Per ambienti multimodali, dove esistono più sequenze di azioni equivalentemente buone, MPPI ha un vantaggio teorico.

Un'ulteriore differenza riguarda la struttura della distribuzione proposta: CEM usa una gaussiana con covarianza aggiornata iterativamente, adattando la direzione di esplorazione alle strutture del paesaggio; MPPI usa una gaussiana con covarianza fissa Σ_ε (o, nelle varianti più avanzate, aggiornata con una stima della varianza delle elite). Questa flessibilità strutturale del CEM nella modellazione della covarianza lo rende preferibile quando il paesaggio del costo ha strutture anisotrope — direzioni lungo cui il costo varia molto più che in altre.

B.7 Confronto con l'ottimizzazione basata sul gradiente

Il metodo alternativo al CEM per la pianificazione latente è l'ottimizzazione diretta tramite discesa del gradiente sulla sequenza di azioni, sfruttando la differenziabilità del rollout del predittore rispetto alle azioni.

Il gradiente della funzione di costo rispetto alla sequenza di azioni $a_{1:H}$ è calcolabile tramite retropropagazione attraverso il rollout:

$$\partial \mathscr{C}/\partial a_t = (\partial \mathscr{C}/\partial \hat{z}_H) \cdot \Pi_{s=t}^{H-1} (\partial \hat{z}_{s+1}/\partial \hat{z}_s) \cdot (\partial \hat{z}_{t+1}/\partial a_t)$$

Il prodotto di Jacobiani $\partial \hat{z}_{s+1}/\partial \hat{z}_s = \nabla_z \text{pred}_\varphi(\hat{z}_s, a_s)$ richiede la retropropagazione attraverso ogni strato del predittore, e il prodotto di H tali Jacobiani può soffrire di gradienti vanishing o exploding per H grande. Per $H = 5$, questo rischio è gestibile, ma aumenta significativamente per $H \geq 10$.

Il confronto operativo tra CEM e discesa del gradiente rivela un trade-off fondamentale. La discesa del gradiente ha complessità per passo $O(H \cdot \text{Costo_backprop})$ e converge a tassi quadratici (con metodi del secondo ordine) o lineari (con discesa del primo ordine); il CEM ha complessità $O(N \cdot T_{iter} \cdot H \cdot \text{Costo_forward})$ e converge a tassi più lenti per la natura stocastica dell'esplorazione, ma è immune ai problemi di gradienti vanishing e non richiede che il rollout sia differenziabile.

I risultati sperimentali riportati nel lavoro su Temporal Straightening di Wang e collaboratori (2026) — il cui contributo verrà analizzato in dettaglio nell'appendice D — mostrano che, quando le traiettorie latenti sono sufficientemente diritte (nel senso della cosine similarity tra velocità latenti consecutive), la discesa del gradiente produce risultati comparabili o superiori al CEM, con un costo computazionale molto inferiore. Questo suggerisce che la scelta tra CEM e ottimizzazione basata sul gradiente non è universalmente determinata dalle proprietà dell'algoritmo, ma dipende criticamente dalla geometria dello spazio latente nel quale la pianificazione viene eseguita: spazi latenti con traiettorie curve rendono difficile la discesa del gradiente (i gradienti puntano in direzioni fuorvianti lontano dai minimi), mentre spazi con traiettorie rette la facilitano, riducendo il vantaggio esplorativo del CEM.

B.8 L'inizializzazione come informazione a priori sull'ottimo

Un aspetto spesso trascurato nell'analisi del CEM è il ruolo dell'inizializzazione della distribuzione proposta come forma implicita di informazione a priori sull'ottimo. In LeWorldModel, la distribuzione iniziale è $\mathcal{N}(0, I)$, il che corrisponde al prior uniforme su azioni piccole intorno all'origine. Se il sistema ha eseguito con successo la pianificazione al passo precedente, il piano ottimale precedente è un'inizializzazione molto più informativa di quella casuale.

Questa osservazione motiva una variante del CEM chiamata *warm-starting*: la distribuzione iniziale al passo t+1 è centrata sulla soluzione ottimale del passo t, eventualmente shifata di una posizione per riflettere l'avanzamento temporale. Formalmente, se $a^*_{1:H}$ è il piano ottimale al passo t, la distribuzione iniziale al passo t+1 è:

$$q_{t+1,0} = \mathcal{N}(a^*_{2:H, 0}, \sigma_{init}^2 I)$$

dove $a^*_{2:H, 0} \in \mathcal{A}^{H}$ *è la sequenza ottenuta traslando il piano precedente di un passo e aggiungendo un'azione nulla in ultima posizione. Il warm-starting riduce tipicamente il numero di iterazioni necessarie per raggiungere la convergenza di un fattore 3−5×, consentendo di utilizzare* T_{iter} più piccolo con lo stesso budget computazionale.

In LeWorldModel, gli autori non esplicitano se sia impiegato il warm-starting. La parametrizzazione con $T_{iter} = 30$ per Push-T (l'ambiente più complesso) suggerisce che non vi sia tale ottimizzazione, poiché con

warm-starting si potrebbe probabilmente usare T_{iter} ≤ 10 con risultati equivalenti.

B.9 Varianti del CEM: iniezione di rumore e la problematica della degenerazione

Una limitazione strutturale del CEM standard è la progressiva degenerazione della distribuzione: man mano che le iterazioni procedono, la varianza stimata della distribuzione diminuisce geometricamente (come mostrato nella sezione B.4), fino a che la distribuzione è così concentrata da campionare solo nella vicinanza immediata della media corrente. Questo impedisce a qualsiasi informazione proveniente da zone diverse del paesaggio di raggiungere l'aggiornamento.

Una soluzione standard è l'*iniezione di rumore* (noise injection o variance floor): invece di aggiornare la varianza con la sola varianza delle elite, si mantiene una varianza minima σ_floor^2:

$$\sigma_{t+1}^2 = \max(\sigma_{t+1,naive}^2, \sigma_floor^2)$$

dove $\sigma_{t+1,naive}^2$ è la varianza stimata dalle elite. Questo previene la degenerazione a spese di una convergenza leggermente più lenta nelle fasi finali. Nella configurazione di LeWorldModel, σ_floor è impostato empiricamente.

Una variante più sofisticata, impiegata in MPPI ma non nel CEM classico, è l'adattamento della temperatura: anziché usare una temperatura τ fissa, si adatta τ in modo da mantenere un certo livello di diversità nella distribuzione delle soluzioni, misurato ad esempio dall'entropia effettiva della distribuzione dei pesi. Questa variante adattiva consente di bilanciare automaticamente esplorazione e sfruttamento nel corso delle iterazioni, senza dover fissare a priori il rapporto K/N.

B.10 Il ruolo del CEM nella struttura duale apprendimento-pianificazione

Da una prospettiva più astratta, l'impiego del CEM nella fase di pianificazione di LeWorldModel è coerente con il principio di separazione tra apprendimento e inferenza che caratterizza l'intera architettura. L'encoder e il predittore,

appresi nella fase offline su $\mathcal{D}$, definiscono una geometria dello spazio latente e una dinamica di transizione che costituiscono il "modello del mondo" stabile. Il CEM, nella fase online, opera come un ottimizzatore che sfrutta questa geometria senza modificarla.

Questa separazione ha una conseguenza importante per l'analisi della convergenza: le garanzie del CEM sono condizionate alla qualità del modello appreso. Se il predittore commette errori sistematici — sovrastimando o sottostimando sistematicamente il costo di certe traiettorie — il CEM convergerà alla soluzione ottimale *rispetto al modello errato*, non rispetto alla funzione di costo vera. Il gap tra l'ottimo del modello e l'ottimo reale è chiamato *model bias* nella letteratura del controllo basato su modelli, ed è una fonte di errore strutturalmente distinta da quella dell'ottimizzazione.

La strategia receding-horizon MPC di LeWorldModel mitiga il model bias attraverso la riplanning frequente: ogni ciclo di pianificazione parte dall'osservazione reale aggiornata, il che consente di correggere i deviazioni accumulate nel ciclo precedente. La qualità della correzione è limitata dall'orizzonte di pianificazione $H = 5$: errori che richiedono più di H passi per essere corretti non vengono rilevati nel ciclo corrente, accumulandosi nel tempo fino a superare la soglia di correggibilità.

Appendice C — Analisi Comparativa Formalizzata di VICReg, PLDM e Strategie Asimmetriche: Limitazioni Teoriche e Fragilità Strutturali

C.1 Premessa: la necessità di una critica formale

La letteratura sui modelli del mondo latenti ha prodotto un insieme di strategie anti-collasso che vengono frequentemente presentate come soluzioni interscambiabili o complementari, senza che le loro limitazioni teoriche siano messe a fuoco con la necessaria precisione. Il confronto qualitativo — "EMA funziona meglio di SG in certi contesti", "VICReg è più stabile di BYOL" — lascia senza risposta le domande che più contano dal punto di vista teorico: quale ipotesi sulla distribuzione dei dati è necessaria perché ciascun metodo garantisca la non-degenerazione? In quale senso le garanzie offerte sono più

deboli delle garanzie di SIGReg? Questa appendice conduce l'analisi nel dettaglio, derivando formalmente le limitazioni strutturali di ciascuna classe di metodi.

C.2 Stop-gradient: struttura del paesaggio di ottimizzazione e non-identificabilità

Lo stop-gradient (SG) interrompe la propagazione del gradiente attraverso il ramo "target" dell'architettura: nella perdita $\|\hat{z}_{t+1} - \mathrm{sg}(z_{t+1})\|^2_2$, il termine $\mathrm{sg}(z_{t+1})$ è trattato come una costante nel calcolo del gradiente, benché dipenda dai parametri θ attraverso $z_{t+1} = \mathrm{enc}_\theta(o_{t+1})$.

Formalmente, la *perdita di predizione con stop-gradient* è:

$$\mathcal{L}_{SG}(\theta, \varphi) = \mathbb{E}[(\|\mathrm{pred}_\varphi(\mathrm{enc}_\theta(o_t), a_t) - \mathrm{sg}(\mathrm{enc}_\theta(o_{t+1}))\|^2_2)]$$

Il gradiente rispetto a θ omette il termine $\partial z_{t+1}/\partial\theta$ che, senza SG, spingerebbe l'encoder a modificare z_{t+1} in modo da ridurre l'errore di predizione — una forza che contrasta il collasso del target. Con SG, questa forza è assente per costruzione.

Teorema (Non-garanzia di non-collasso con SG): Esiste una distribuzione di dati $\mathcal{D}$ e un'architettura (encoder, predittore) tali che la minimizzazione di $\mathcal{L}_{SG}$ converge alla soluzione collassata $\mathrm{enc}_\theta(o) = c$ per ogni $o \in \mathcal{O}$.

Dimostrazione: Si considera il dataset con una sola coppia di stati distinti (o_0, o_1) ripetuta. Per $\mathrm{enc}_\theta(o) = c$ (costante), $\mathcal{L}_{SG} = \|\mathrm{pred}_\varphi(c, a) - c\|^2_2$, minimizzata da $\mathrm{pred}_\varphi(c, a) = c$ per ogni a. Il gradiente rispetto a θ è zero (sg non dipende da θ nell'aggiornamento) e rispetto a φ è zero (soluzione $\mathrm{pred} = c$ è ottima). La coppia ($\mathrm{enc} = c$, $\mathrm{pred} = c$) è quindi un punto stazionario di $\mathcal{L}_{SG}$ con perdita nulla — un punto fisso dell'algoritmo di discesa del gradiente. □

Questo risultato mostra che lo SG da solo non impedisce il collasso: la soluzione costante è sempre un punto stazionario della perdita con SG, indipendentemente dalla struttura dei dati.

C.3 Medie mobili esponenziali: analisi delle orbite e instabilità dell'aggiornamento target

L'Exponential Moving Average (EMA) combina lo SG con l'aggiornamento lento del secondo encoder ("target encoder"):

$\bar{\theta}_t = \tau\,\bar{\theta}_{t-1} + (1-\tau)\,\theta_t,\ \tau \in (0, 1)$

L'analisi formale del sistema EMA nel contesto JEPA richiede la costruzione di un sistema dinamico con stato $(\theta_t, \bar{\theta}_t)$:

$\theta_t = \theta_{t-1} - \alpha\,\nabla_\theta \mathcal{L}_{JEPA}(\theta_{t-1}, \bar{\theta}_{t-1})\ \bar{\theta}_t = \tau\,\bar{\theta}_{t-1} + (1-\tau)\,\theta_t$

Il punto fisso del sistema: I punti fissi sono le coppie $(\theta^*, \bar{\theta}^*)$ tali che $\theta^* = \bar{\theta}^*$ e $\nabla_\theta\,\mathcal{L}_{JEPA}(\theta^*, \theta^*) = 0$. Sostituendo $\bar{\theta} = \theta$ nella perdita si ottiene esattamente $\mathcal{L}_{SG}$. I punti fissi del sistema EMA coincidono dunque con i punti stazionari di $\mathcal{L}_{SG}$ — e quindi la soluzione collassata è ancora un punto fisso.

La domanda rilevante è se il sistema EMA-SG converga *di fatto* al collasso. La risposta dipende dalla struttura della matrice Jacobiana del sistema attorno alla soluzione collassata:

$J_{collasso} = [\partial^2\mathcal{L}_{SG}/\partial\theta^2,\ (1-\tau)\cdot I;\ (1-\tau)\cdot I,\ \tau\cdot I]$

Se tutti gli autovalori di $J_{collasso}$ hanno parte reale negativa, la soluzione collassata è localmente stabile. Analisi numeriche su architetture di piccola scala mostrano che, per τ vicino a 1 e learning rate α piccolo, $J_{collasso}$ ha autovalori con parte reale sia positiva che negativa, indicando una varietà instabile: il sistema può sfuggire al collasso se perturbato nella direzione degli autovalori instabili, ma converge al collasso se perturbato nella direzione di quelli stabili. La pratica empirica di scegliere $\tau \in [0.99, 0.999]$ agisce su questa struttura spettrale, rendendo più ampia la varietà instabile, senza però eliminarla del tutto.

C.4 VICReg: incompletezza della garanzia e collasso non rilevato

VICReg impone tre vincoli sulla distribuzione degli embedding:

$\mathcal{L}_{VICReg}(\theta, \varphi) = \alpha\,\mathcal{L}_{inv} + \beta\,\mathcal{L}_{var} + \gamma\,\mathcal{L}_{cov}$

L'analisi delle limitazioni richiede di esaminare ciascun termine separatamente.

Limitazione del termine di varianza: $\mathcal{L}_{var} = (1/d) \Sigma_d \max(0, 1 - \text{Std}(z^{(d)}))$ penalizza ogni componente con deviazione standard sotto 1. Questa penalità non previene il collasso parziale: un encoder che produce embedding con varianza unitaria in tutte le d dimensioni ma concentrati su una sottovarietà k-dimensionale ($k \ll d$) soddisfa $\mathcal{L}_{var} = 0$ pur essendo degenere in senso informativo.

Esempio formale di collasso non rilevato: Si consideri l'encoder $z = [v_1(o) + \varepsilon_1, v_1(o) + \varepsilon_2, ..., v_1(o) + \varepsilon_d]$ dove v_1 è la prima componente principale dei dati e $\varepsilon_j \sim \mathcal{N}(0, 0.01)$. Ogni componente ha varianza $\approx \text{Var}(v_1) + 0.01 > 0$, dunque $\mathcal{L}_{var} = 0$. Eppure gli embedding sono concentrati sulla diagonale principale di $\mathbb{R}^d$ — una varietà monodimensionale — con rango effettivo $\approx 1 \ll d$. In questo caso, le covarianze tra $z^{(j)}$ e $z^{(k)}$ sono $\text{Cov}(v_1 + \varepsilon_j, v_1 + \varepsilon_k) \approx \text{Var}(v_1) \gg 0$, quindi $\mathcal{L}_{cov} \gg 0$ rileverebbe il problema. Ma se le dimensioni extra sono generate da componenti principali ortogonali con varianza unitaria anziché rumore correlato, si ottiene $z = [v_1(o), v_2(o), ..., v_k(o), e_{k+1}, ..., e_d]$ dove $e_j \sim \mathcal{N}(0, 1)$ indipendenti. In questo caso: varianze unitarie ($\mathcal{L}_{var} = 0$), covarianze zero per ortogonalità ($\mathcal{L}_{cov} \approx 0$), ma la distribuzione ha rango effettivo k per le componenti informative e rango effettivo d per le componenti rumorose — un embedding parzialmente degenerato che VICReg non rileva.

Limitazione del termine di covarianza: $\mathcal{L}_{cov} = (1/d) \Sigma_{j \neq k} \text{Cov}(z^{(j)}, z^{(k)})^2$ penalizza le correlazioni pairwise tra componenti. Questa condizione è necessaria ma non sufficiente per la gaussianità: una distribuzione può avere covarianza zero per tutti i pari (j, k) pur non essendo gaussiana — ad esempio una distribuzione laplaciana o una distribuzione su una varietà non-lineare decorrelata. Il termine $\mathcal{L}_{cov} = 0$ garantisce l'assenza di dipendenze lineari tra componenti, ma non l'assenza di dipendenze non-lineari, che sono perfettamente compatibili con valori di $\mathcal{L}_{cov}$ nulli.

La conclusione formale è che VICReg non garantisce che la distribuzione degli embedding abbia dimensione effettiva pari a d: garantisce solo che le varianze marginali superino una soglia e che le covarianze a coppie siano piccole — condizioni necessarie ma non sufficienti per la piena non-degenerazione.

C.5 PLDM: analisi delle instabilità multi-termine

PLDM produce l'obiettivo con sette termini e sei scalari liberi:

$\mathcal{L}_{PLDM} = \mathcal{L}_{pred} + \alpha \mathcal{L}_{var} + \beta \mathcal{L}_{cov} + \gamma \mathcal{L}_{time\text{-}sim} + \zeta \mathcal{L}_{time\text{-}var} + \nu \mathcal{L}_{time\text{-}cov} + \mu \mathcal{L}_{IDM}$

Conflitto strutturale tra $\mathcal{L}_{time\text{-}sim}$ e $\mathcal{L}_{time\text{-}var}$: Il termine di similarità temporale $\mathcal{L}_{time\text{-}sim} = (1/(BT)) \Sigma_{i,t} \|z_t^{(i)} - z_{t+1}^{(i)}\|^2_2$ incentiva la smoothness delle traiettorie. Il termine di varianza temporale $\mathcal{L}_{time\text{-}var}$ incentiva che ogni traiettoria spazzi una porzione non trascurabile dello spazio. Per una traiettoria con moto quasi-uniforme $z_t \approx z_0 + vt$:

$\mathcal{L}_{time\text{-}sim} \approx \|v\|^2/T + \sigma_\varepsilon^2 \quad \mathcal{L}_{time\text{-}var} \approx \max(0, 1 - \|v\|\sqrt{T})$

Ridurre $\|v\|$ (embedding lentamente variabili) abbassa $\mathcal{L}_{time\text{-}sim}$ ma aumenta $\mathcal{L}_{time\text{-}var}$ quando $\|v\|\sqrt{T} < 1$. Il punto di equilibrio dipende dal rapporto γ/ζ: se $\gamma \gg \zeta$, le traiettorie sono quasi statiche; se $\zeta \gg \gamma$, sono rumorose. Ogni ambiente richiederebbe una taratura diversa di questo rapporto, spiegando la necessità di iperparametri ambiente-specifici osservata in PLDM.

Conflitto tra $\mathcal{L}_{IDM}$ e $\mathcal{L}_{pred}$: Il termine della dinamica inversa $\mathcal{L}_{IDM} = \|g_\psi(z_t, z_{t+1}) - a_t\|^2_2$ impone che coppie con azioni diverse vengano mappate in embedding diversamente distanziati. Questo vincolo è però orientato alla distinzione locale delle transizioni, non alla qualità globale della rappresentazione: un encoder che separa adeguatamente le coppie (z_t, z_{t+1}) per diversi a_t può comunque produrre rappresentazioni di qualità insufficiente per la pianificazione su orizzonti più lunghi.

Sensibilità agli iperparametri: Il numero di valutazioni necessario per trovare la combinazione $(\alpha, \beta, \gamma, \zeta, \nu, \mu)$ soddisfacente cresce come n^6 per una griglia di n valori per parametro. Per $n = 5$ questo è 15.625 addestramenti; per $n = 10$, è 10^6. In pratica, metodi bayesiani riducono il costo a 100−300 valutazioni, ma questo rappresenta comunque un overhead di 100× rispetto al costo di un singolo addestramento, non computato nel confronto sperimentale con LeWorldModel.

C.6 Analisi della stabilità: il gap di riproducibilità

Si definisce il *gap di stabilità* come misura della riproducibilità empirica di un metodo:

Gap_M = 2σ_M/√K

dove σ_M è la deviazione standard del tasso di successo su K repliche indipendenti. Per K = 3 repliche:

Gap_{LeWM} = 2 · 2.83/√3 ≈ 3.27 punti percentuali Gap_{PLDM} = 2 · 5.00/√3 ≈ 5.77 punti percentuali Gap_{DINO-WM} = 2 · 1.63/√3 ≈ 1.88 punti percentuali

Il gap di DINO-WM (1.88) è inferiore a quello di LeWorldModel (3.27), a indicare che la stabilità dell'encoder congelato produce addestramento più riproducibile — conseguenza naturale del fatto che i parametri dell'encoder non vengono ottimizzati, eliminando una fonte primaria di variabilità. Il gap di PLDM (5.77) è quasi il doppio di quello di LeWorldModel, a testimonianza della fragilità introdotta dalla molteplicità dei termini di perdita.

C.7 La gerarchia formale delle garanzie anti-collasso

Sulla base dell'analisi condotta nelle sezioni precedenti, si può costruire una gerarchia formale delle garanzie offerte dai diversi metodi.

Il Livello 0, senza alcuna garanzia, corrisponde allo SG isolato: la soluzione collassata è sempre un punto stazionario della perdita; la non-degenerazione dipende interamente dall'inizializzazione e dalla dinamica stocastica dell'ottimizzazione, senza alcun fondamento formale.

Il Livello 1, con garanzia contingente, corrisponde a EMA-SG, BYOL, SimSiam: la soluzione collassata può essere resa instabile attraverso la scelta appropriata di τ, ma la garanzia dipende dal regime operativo e non è derivabile dall'obiettivo ottimizzato come proprietà strutturale.

Il Livello 2, con garanzia statistica parziale, corrisponde a VICReg e PLDM: condizioni necessarie ma non sufficienti per la non-degenerazione sono formalmente incorporate nell'obiettivo; il collasso viene rilevato con alta probabilità per distribuzioni significativamente concentrate, ma esistono soluzioni parzialmente degeneri che soddisfano i vincoli con perdita nulla.

Il Livello 3, con garanzia distribuzionale completa, corrisponde a SIGReg: la distribuzione target è specificata esplicitamente come $\mathcal{N}$(0, I_d); la statistica di

test è nulla se e solo se la distribuzione degli embedding coincide con il target — per il teorema di Cramér-Wold nel limite M → ∞; qualsiasi forma di collasso produce valori positivi della statistica indipendentemente dalla struttura specifica della degenerazione.

La posizione di SIGReg al Livello 3 corrisponde a una distinzione qualitativa, non solo quantitativa, rispetto ai metodi dei livelli precedenti: mentre EMA-SG e VICReg operano attraverso incentivi impliciti alla diversità, SIGReg impone esplicitamente il matching con una distribuzione di non-collasso, rendendo la garanzia derivabile direttamente dalla struttura dell'obiettivo ottimizzato.

C.8 Encoder pre-addestrati: la scelta per elusione e i suoi costi

La strategia degli encoder pre-addestrati (DINOv2 in DINO-WM) si colloca al di fuori della gerarchia precedente: essa non previene il collasso ma lo rende irrilevante per l'encoder, spostando tutta la pressione anti-collasso sul pre-training. Dal punto di vista formale, questa è una forma di prior implicito sulla distribuzione degli embedding: il pre-training su 124 milioni di immagini naturali genera una distribuzione P_Z^{DINOv2} che è empiricamente non-collassata per la varietà del dataset di pre-training.

Il costo formale di questa strategia è la perdita di adattabilità distribuzionale. Si definisce il *gap di adattamento* come:

Gap_{adapt}(DINO) = Performance(P_Z^{opt}) − Performance(P_Z^{DINOv2})

dove P_Z^{opt} è la distribuzione ottimale degli embedding per la pianificazione nell'ambiente considerato. Per gli ambienti robotici 2D (Wall, PushT, UMaze), il gap è positivo — LeWorldModel supera DINO-WM — indicando che P_Z^{opt} diverge sufficientemente da P_Z^{DINOv2} da rendere vantaggioso l'addestramento specifico del dominio. Per OGBench-Cube 3D, il gap è negativo: la ricchezza del pre-training DINOv2 non è colmabile con il dataset limitato di 10.000 episodi disponibile per l'addestramento specifico.

Questa dipendenza dal volume del dataset suggerisce che il gap di adattamento è una funzione decrescente dell'ammontare di dati dominio-specifici disponibili: nel limite di dati abbondanti, l'addestramento end-to-end

convergerebbe verso P_Z^{opt}, superando strutturalmente il pre-training; nel limite di dati scarsi, il pre-training mantiene un vantaggio strutturale non colmabile.

C.9 Implicazioni per la progettazione di architetture JEPA future

L'analisi condotta in questa appendice suggerisce tre direzioni progettuali per superare le limitazioni identificate. La prima è l'estensione di SIGReg a distribuzioni target più flessibili con garanzie di Livello 3 mantenute. Distribuzioni di mixture gaussiane con K componenti — adatte a spazi degli stati multimodali come Two-Room — hanno funzioni caratteristiche della forma $\Sigma_k \pi_k \exp(it(u^\top \mu_k) - \sigma_k^2 t^2/2)$ calcolabili in O(K) operazioni, il che rende questa estensione computazionalmente accessibile anche per $K = 20$ componenti. Il costo è l'introduzione di $K - 1$ parametri aggiuntivi per le medie delle componenti, una fonte di iper-parametrizzazione che deve essere gestita con criteri bayesiani o di tipo model selection.

La seconda direzione riguarda la garanzia di non-collasso condizionata all'azione. Le strategie analizzate si concentrano sulla distribuzione marginale degli embedding, non su quella condizionata all'azione. Un encoder potrebbe produrre embedding gaussiani isotropici marginalmente ma con struttura condizionale concentrata: z_{t+1} quasi determinato dalla coppia (z_t, a_t). Una variante di SIGReg che valuti la gaussianità dei residui $z_{t+1} - \text{pred}_\varphi(z_t, a_t)$ anziché di z_{t+1} affronterebbe questa limitazione, imponendo che l'*informazione non predetta* dall'azione sia distribuita gaussianamente — una forma di bottleneck informazionale che separa la struttura predittiva da quella residuale.

La terza direzione è la formalizzazione del compromesso tra anti-collasso e geometria latente favorevole alla pianificazione, che costituisce il tema centrale della sezione dedicata al temporal straightening nella continuazione di questa monografia.

C.10 Caso empirico: l'anomalia del ResNet come collasso direzionale non rilevato da VICReg

La sezione 5.6 della monografia registra che il ResNet-18, impiegato come encoder al posto del ViT-T' predefinito, produce prestazioni competitive (94.0% vs 96.0% su Push-T) confermando la robustezza di LeWorldModel rispetto alla famiglia architettonica dell'encoder. Questa osservazione è corretta ma incompleta: il dato di Wang et al. (2026) aggiunge una dimensione che modifica sensibilmente la lettura dell'esperimento. Senza raddrizzamento esplicito, il ResNet addestrato da zero produce curvature latenti abnormalmente elevate e performance di pianificazione vicine allo zero sugli ambienti Wall e PointMaze — risultati peggiori perfino di quelli ottenuti con un DINOv2 congelato, che non viene aggiornato in alcun modo durante l'addestramento specifico del dominio. Con raddrizzamento esplicito, lo stesso ResNet diventa sistematicamente l'encoder con le migliori prestazioni su tutti gli ambienti considerati.

Questo fenomeno è teoricamente significativo per le ragioni esposte nell'analisi di VICReg condotta nella sezione C.4. Un encoder ResNet addestrato da zero ha piena libertà di adattare la propria geometria latente ai dati di addestramento: a differenza di un backbone congelato, non porta con sé alcun vincolo geometrico derivante dal pre-training. Questa flessibilità è una risorsa — consente di apprendere rappresentazioni specifiche del dominio — ma è anche una fonte di rischio: in assenza di un vincolo geometrico esplicito sulle traiettorie, l'encoder può convergere verso soluzioni in cui la distribuzione marginale degli embedding è gaussiana e isotropica (soddisfacendo formalmente VICReg e SIGReg) ma le transizioni consecutive puntano in direzioni arbitrarie, producendo curvature latenti elevate. Si tratta precisamente della forma di collasso che la sezione C.4 definisce come non rilevata da VICReg: il termine di varianza garantisce che nessuna singola coordinata sia costante, il termine di covarianza garantisce la decorrelazione, ma nessuno dei due termini pone vincoli sulla struttura temporale direzionale degli embedding — ossia su come la direzione del vettore di velocità latente $v_t := z_{t+1} - z_t$ evolve da un passo al successivo.

Il caso ResNet illustra così un punto cieco strutturale delle metriche di anti-collasso basate sulla distribuzione marginale: esse garantiscono la diversità degli embedding istante per istante, ma non la coerenza della geometria delle transizioni nel tempo. Un encoder che produca embedding uniformemente distribuiti ma con velocità latenti orientate casualmente supera i test di VICReg e di SIGReg sulla distribuzione marginale, ma genera un paesaggio di perdita per il pianificatore con condizionamento del Gramiano elevato — esattamente la situazione descritta dalla disuguaglianza (7.11) con ε grande. Il DINOv2

congelato, in questo confronto, si trova in una posizione paradossalmente favorevole: la geometria delle sue rappresentazioni è stata modellata da 124 milioni di immagini naturali, e la struttura delle transizioni nel dataset robotico 2D, pur non essendo quella per cui il backbone è stato addestrato, non è sufficientemente aliena da produrre curvature catastrofiche. Il ResNet, partendo da zero sullo stesso dataset limitato e senza alcun vincolo direzionale, non ha questo vantaggio e collassa in una configurazione geometricamente incoerente.

Il risultato del raddrizzamento esplicito completa il quadro e ne inverte il segno: imponendo che la cosine similarity tra velocità latenti consecutive sia elevata, il vincolo di curvatura trasforma la flessibilità del ResNet da rischio a vantaggio. Con ε mantenuto basso dalla regolarizzazione, l'encoder libero di adattarsi al dominio produce le traiettorie più rettilinee e le performance di pianificazione più elevate — esattamente l'opposto del comportamento senza raddrizzamento. Questo caso empirico illustra con precisione il tema centrale dell'addestramento end-to-end che attraversa l'intera monografia: la flessibilità dell'encoder è una risorsa condizionata alla disponibilità di vincoli geometrici adeguati. In loro assenza, il rischio di collasso direzionale — una forma di degenerazione che le metriche di anti-collasso distribuzionale non rilevano — è reale e quantificabile. In loro presenza, lo stesso encoder che collassa diventa il più performante.

Capitolo 7. Raddrizzamento Temporale, Teoria del Controllo Ottimale e Collo di Bottiglia Informazionale

7.1 La geometria dello spazio latente come problema di controllo

Il nesso tra rappresentazione latente e pianificazione non è soltanto una questione di efficienza algoritmica: esso tocca qualcosa di più profondo, che riguarda la struttura matematica dello spazio entro cui si svolge l'ottimizzazione. Quando si formula il problema della pianificazione come ricerca di una sequenza di azioni che minimizza la distanza tra lo stato terminale predetto e un obiettivo specificato nello spazio latente, ci si trova a operare su una funzione obiettivo la cui geometria — convessità, curvatura,

condizionamento del Hessiano — dipende in modo critico dalla geometria dello stesso spazio di rappresentazione. Questo punto è concettualmente centrale e vale la pena di svilupparlo con la precisione che merita.

La teoria classica del controllo ottimale, inaugurata dai lavori di Pontryagin e dai suoi collaboratori negli anni Cinquanta del Novecento e sistematizzata da Kalman, Bucy e successori nel decennio seguente, ha sempre riconosciuto che le proprietà geometriche dello spazio degli stati sono determinanti per la struttura delle soluzioni ottimali. In particolare, la raggiungibilità di uno stato obiettivo — ossia l'esistenza di un ingresso di controllo che trasferisca il sistema dallo stato iniziale a quello finale — è codificata nel Gramiano di controllabilità, un oggetto algebrico che concentra in una singola matrice l'intera dipendenza del problema dal modello dinamico e dall'asse temporale della pianificazione. La connessione tra la geometria delle traiettorie latenti e questo apparato teorico — apparentemente distante — costituisce il filo conduttore della presente sezione.

Si consideri il modello del mondo latente nella sua forma più elementare, quella di un sistema dinamico lineare tempo-invariante in spazio degli stati. Formalmente, sia $z_t \in \mathbb{R}^d$ il vettore di stato latente al tempo t e $a_t \in \mathbb{R}^{d_a}$ il vettore di azione al medesimo istante. La dinamica è descritta dalla relazione di ricorrenza

$$z_{t+1} = A\, z_t + B\, a_t , \qquad A \in \mathbb{R}^{d \times d},\ B \in \mathbb{R}^{d \times d_a} . \tag{7.1}$$

Questa ipotesi di linearità non è un'assunzione realistica circa il comportamento di ambienti fisici complessi, ma costituisce il quadro di analisi entro cui è possibile derivare garanzie teoriche precise e interpretare i risultati empirici come proiezioni locali di fenomeni più generali. Nel seguito si discuterà in che misura i risultati ottenuti in questo contesto si estendono a predittori non lineari.

L'obiettivo della pianificazione è la ricerca di una sequenza di azioni $\mathbf{a} = (a_0, a_1, \dots, a_{K-1}) \in \mathbb{R}^{K \times d_a}$ che minimizzi la perdita terminale

$$\mathcal{L}(\mathbf{a}) = \lVert z_K - z_g \rVert_2^2 , \quad z_K = \Phi(\mathbf{a}) , \tag{7.2}$$

dove Φ denota lo srotolamento della dinamica (7.1) a partire dallo stato iniziale fisso z_0 per un orizzonte di K passi e z_g è lo stato obiettivo. Lo srotolamento produce la mappa affine

$$z_K = A^K z_0 + \Sigma_{t=0}^{K-1} A^{K-1-t} B a_t , \tag{7.3}$$

che è lineare in $\mathbf{a}$ una volta fissato z_0. Poiché la perdita (7.2) è il quadrato della norma di una funzione affine, essa è una forma quadratica convessa in $\mathbf{a}$, e il suo Hessiano — matrice delle derivate seconde rispetto alle azioni — è costante, indipendente dal punto di valutazione:

$$H := \nabla^2_{\mathbf{a}} \mathcal{L}(\mathbf{a}) = 2 J_\Phi^\top J_\Phi \succeq 0 , \tag{7.4}$$

dove $J_\Phi := \partial z_K / \partial \mathbf{a} = [A^{K-1}B, A^{K-2}B, \dots, B] \in \mathbb{R}^{d \times K d_a}$ è lo Jacobiano della mappa di srotolamento rispetto all'intera sequenza di azioni.

La struttura del Hessiano è dunque interamente determinata da J_Φ, che a sua volta dipende dalle potenze della matrice di transizione A e dalla matrice di ingresso B. Il condizionamento di H — il rapporto tra il valore singolare massimo e quello minimo non nullo — governa la velocità di convergenza del gradiente discendente applicato alla perdita di pianificazione. Un Hessiano mal condizionato produce una superficie di perdita con valli elongate in alcune direzioni e ripide in altre, rendendo necessari passi di apprendimento molto piccoli o l'impiego di metodi del secondo ordine per garantire la convergenza.

7.2 Il Gramiano di controllabilità come misura di raggiungibilità

Il prodotto $J_\Phi J_\Phi^\top \in \mathbb{R}^{d \times d}$ ha una denominazione propria e un'interpretazione profonda nella teoria dei sistemi lineari. Esso è il Gramiano di controllabilità a orizzonte finito, abitualmente denotato W_K:

$$W_K := J_\Phi J_\Phi^\top = \Sigma_{k=0}^{K-1} A^k B B^\top (A^\top)^k . \tag{7.5}$$

Il Gramiano W_K codifica, in forma compatta, l'energia minima richiesta per trasferire il sistema dallo stato zero a un qualunque stato target nell'orizzonte K. Più precisamente, per ogni stato target z_g nello spazio immagine di J_Φ, l'energia minima della sequenza di controllo che raggiunge z_g è data da

$$\mathcal{E}^{**}(z_g) = z_g^{\top} W_K^{-1} z_g , \qquad (7.6)$$

formula che vale quando W_K è invertibile, ossia quando il sistema è completamente controllabile nell'orizzonte considerato. Nel caso in cui d_a < d, la matrice B non è invertibile e W_K può essere singolare al di fuori del sottospazio controllabile $\mathscr{S}_K := \mathrm{im}(W_K)$; in tal caso l'espressione (7.6) si restringe a $\mathscr{S}_K$ con l'impiego dello pseudoinverso.

La connessione con il problema di pianificazione è diretta: poiché i valori singolari non nulli di $J_\Phi^{\top} J_\Phi$ coincidono con quelli di $J_\Phi J_\Phi^{\top} = W_K$, il numero di condizionamento effettivo del Hessiano H soddisfa

$$\varkappa_{\mathrm{eff}}(H) = \varkappa(W_K) := \sigma_{\max}(W_K) / \sigma^{+}_{\min}(W_K) , \qquad (7.7)$$

dove $\sigma^{+}_{\min}$ indica il valore singolare minimo non nullo. Dunque il condizionamento del problema di ottimizzazione è esattamente il condizionamento del Gramiano di controllabilità: un Gramiano ben condizionato è equivalente, nel contesto della pianificazione differenziabile, a una funzione obiettivo il cui paesaggio è quasi isotropo, favorevole alla convergenza di qualunque metodo basato sul gradiente.

Questo risultato stabilisce un ponte formale tra una nozione classica della teoria dei sistemi — il condizionamento del Gramiano come misura della difficoltà del controllo — e una nozione emergente dell'apprendimento automatico — il condizionamento del Hessiano come misura della difficoltà dell'ottimizzazione per la pianificazione. Non si tratta di un'analogia superficiale, bensì di un'identità algebrica: i due concetti, sviluppati in tradizioni disciplinari distinte, si rivelano descrizioni matematicamente equivalenti dello stesso fenomeno.

7.3 Transizioni ε-rette e analisi della controllabilità perturbata

Si introduce ora la nozione cruciale che collega la geometria delle traiettorie latenti con le proprietà del Gramiano. Nella letteratura dei sistemi lineari, la matrice A = I rappresenta il caso di dinamica puramente traslazionale: lo stato successivo è identico allo stato corrente a meno del termine di controllo Ba_t. Questa situazione corrisponde a traiettorie latenti esattamente rettilinee in assenza di controllo, nel senso che la componente libera dell'evoluzione (omogenea) z_{t+1} = A z_t produce incrementi z_{t+1} - z_t = (A - I)z_t che si annullano quando A = I.

Più in generale, si definisce una transizione ε-retta come una dinamica lineare in cui

$$\|A - I\|_2 \le \varepsilon \, . \tag{7.8}$$

La condizione (7.8) impone che la norma spettrale della perturbazione di A rispetto all'identità sia al più ε. Per il teorema di perturbazione di Weyl, i valori singolari di A soddisfano allora $1 - \varepsilon \le \sigma_{\min}(A) \le \sigma_{\max}(A) \le 1 + \varepsilon$. Di conseguenza, il numero di condizionamento di A è

$$\kappa(A) = \sigma_{\max}(A)/\sigma_{\min}(A) \le (1 + \varepsilon)/(1 - \varepsilon) \, , \tag{7.9}$$

e per ε piccolo questo è prossimo all'unità. Inserendo questo bound nella stima del condizionamento del Gramiano si ottiene

$$\kappa_{\mathrm{eff}}(H) \le \kappa(B)^2 \cdot ((1 + \varepsilon)/(1 - \varepsilon))^{2(K-1)} \, . \tag{7.10}$$

Per $\varepsilon \le 1/2$ si può ulteriormente maggiorare il membro destro usando la stima $\ln((1+\varepsilon)/(1-\varepsilon)) \le 3\varepsilon$ valida in tale intervallo, che conduce alla forma esponenziale

$$\kappa_{\mathrm{eff}}(H) \le \kappa(B)^2 \cdot e^{6\varepsilon K} \, . \tag{7.11}$$

La disuguaglianza (7.11) è di notevole valore interpretativo. Essa afferma che, per un sistema con transizioni sufficientemente vicine all'identità — ossia per ε piccolo — il numero di condizionamento del Hessiano di pianificazione cresce al più esponenzialmente in εK, non in K. In altre parole, la rettitudine delle traiettorie agisce come un moltiplicatore dell'orizzonte effettivo: un sistema con ε = 0 ha condizionamento costante pari a $\varkappa(B)^2$ indipendentemente dall'orizzonte; all'aumentare di ε il condizionamento si degrada, ma il tasso di degradazione è controllato dall'ampiezza della perturbazione rispetto all'identità.

L'implication pratica è altrettanto limpida. Un encoder che produca traiettorie latenti più rettilinee — nel senso che i vettori di velocità latente v_t := z_{t+1} − z_t mantengono la stessa direzione lungo la traiettoria — corrisponde, nella dinamica lineare approssimante, a una matrice A più vicina all'identità, e dunque a un condizionamento più favorevole del problema di pianificazione. La regolarizzazione di curvatura introdotta dal temporal straightening si traduce, pertanto, in un miglioramento diretto e quantificabile delle proprietà di convergenza del pianificatore a gradiente.

L'evidenza empirica fornita dalla Tabella 1 e dalla Tabella 3 di Wang et al. (2026) quantifica con precisione il vantaggio predetto dall'analisi teorica. Il risultato più significativo riguarda l'ambiente UMaze: senza raddrizzamento esplicito, il gradiente discendente in modalità open-loop raggiunge un tasso di successo del 44%; con raddrizzamento esplicito, lo stesso pianificatore a gradiente sale al 94% — un incremento di cinquanta punti percentuali che lo porta sostanzialmente alla parità con il controllore MPC chiuso. Questo salto non è marginale: esso segnala che la condizione di quasi-identità per A, formalmente motivata dalla disuguaglianza (7.11), si traduce in un regime qualitativamente diverso del paesaggio di perdita, non semplicemente in un miglioramento quantitativo del condizionamento.

Il confronto sistematico tra gradiente discendente (GD) e stima dell'entropia crociata (CEM) riportato nella Tabella 3 di Wang et al. (2026) completa il quadro empirico. Su alcuni ambienti — in particolare UMaze e PointMaze — il divario tra GD e CEM si azzera completamente in presenza di raddrizzamento esplicito, mentre rimane sostanziale in sua assenza. Questo risultato è la conferma diretta del meccanismo teorizzato nella sezione corrente: il raddrizzamento, abbassando il numero di condizionamento del Gramiano

W_K secondo il bound (7.11), rende il paesaggio di perdita sufficientemente regolare da consentire al gradiente discendente — un metodo del primo ordine — di competere con la ricerca stocastica di secondo ordine implicita nel CEM. Su ambienti a curvatura intrinsecamente alta come Wall, il divario persiste, confermando che la garanzia (7.11) è legata al grado di raddrizzamento effettivamente raggiunto e non costituisce una promessa incondizionata.

7.4 La curvatura latente come proxy della rettitudine della dinamica

Il collegamento tra la cosine similarity tra velocità latenti consecutive — la grandezza ottimizzata dalla regolarizzazione di curvatura — e il parametro ε di rettitudine richiede un argomento che va enunciato con cura, poiché esso non è immediato ma dipende da ipotesi ragionevoli sulle proprietà della traiettoria.

Si assumano velocità latenti di norma costante, ossia $\|v_t\|_2 = c$ per ogni t, e azioni con variazione limitata $\Delta a := \max_t \|a_{t+1} - a_t\|_2 < \infty$. Sotto queste ipotesi, la differenza tra velocità consecutive è

$$\begin{aligned} v_{t+1} - v_t &= (z_{t+2} - z_{t+1}) - (z_{t+1} - z_t) \\ &= A(z_{t+1} - z_t) + B(a_{t+1} - a_t) - (z_{t+1} - z_t) \\ &= (A - I)v_t + B(a_{t+1} - a_t) \,. \end{aligned} \tag{7.12}$$

Per la disuguaglianza triangolare si ha dunque

$$\|(A - I)\,\hat{v}_t\|_2 \le \|v_{t+1} - v_t\|_2 / c + \sigma_{\max}(B)\,\Delta a / c \,, \tag{7.13}$$

dove $\hat{v}_t := v_t / \|v_t\|_2$ è il versore della velocità latente al tempo t. Poiché $\|v_{t+1} - v_t\|_2^2 = 2c^2(1 - \mathscr{C}_t)$ con $\mathscr{C}_t$ la cosine similarity tra v_t e v_{t+1}, si ottiene $\|v_{t+1} - v_t\|_2 = c\sqrt{2(1 - \mathscr{C}_t)}$ e quindi

$$\|(A - I)\,\hat{v}_t\|_2 \le \sqrt{2(1 - \mathscr{C}_t)} + \sigma_{\max}(B)\,\Delta a / c \,. \tag{7.14}$$

La (7.14) è la stima fondamentale: essa collega la deviazione di A dall'identità lungo le direzioni visitate — le direzioni dei versori di velocità v̂_t — con la curvatura delle traiettorie latenti misurata dalla cosine similarity. In particolare, se la cosine similarity media $\bar{\mathscr{C}} := (K-1)^{-1} \Sigma_{t=0}^{K-2} \mathscr{C}_t$ è almeno $1 - \eta$, allora per la concavità di $\sqrt{(\cdot)}$ si ottiene

$$(K-1)^{-1} \Sigma_{t=0}^{K-2} \|(A - I) \hat{v}_t\|_2 \leq \sqrt{(2\eta)} + \sigma_{max}(B) \Delta a / c . \quad (7.15)$$

La (7.15) mostra che massimizzare la cosine similarity media lungo le traiettorie di addestramento è un modo di approssimare la condizione $\varepsilon \approx 0$, nella misura in cui le direzioni visitate ricoprono in modo sufficiente lo spazio latente. Naturalmente, questo controllo è direzionale: esso vincola $\|(A - I)\hat{v}_t\|_2$ per le direzioni v̂_t effettivamente percorse durante l'addestramento, non l'intera norma spettrale di $(A - I)$. Il passaggio da un bound direzionale a uno spettrale richiede una condizione di copertura, ossia l'assunzione che l'insieme dei versori $\{\hat{v}_t\}$ sia sufficientemente denso nello spazio $\mathbb{R}^d$. Nelle situazioni in cui i dati di addestramento sono raccolti con sufficiente diversità, questa condizione è praticabile.

7.5 Dalla regolarizzazione di curvatura al paesaggio di perdita quasi-convesso

La questione delle garanzie teoriche sulla convergenza si arricchisce di un aspetto ulteriore quando si passa dal regime lineare a quello non lineare. Per un predittore non lineare f_θ, la mappa $z_K = \Phi(\mathbf{a})$ non è più affine e lo Hessiano di $\mathcal{L}$ dipende in generale dal punto **a** corrente. In tale contesto, la convergenza del gradiente discendente non è assicurata da proprietà del Hessiano in un punto, ma dalla curvatura globale della funzione obiettivo.

Ciononostante, il temporal straightening produce un effetto documentato empiricamente anche nel regime non lineare: il paesaggio della perdita di pianificazione nello spazio delle azioni si avvicina a una configurazione quasi-convessa. Questo fenomeno, illustrato attraverso visualizzazioni bidimensionali della funzione valore minimo in funzione della prima azione dell'orizzonte (con le azioni successive ottimizzate a parità di prima azione), mostra come la superficie di perdita prodotta da un encoder non raddrizzato

presenti molteplici minimi locali pronunciati, mentre quella prodotta da un encoder sottoposto a temporal straightening sia significativamente più unimodale e regolare.

L'interpretazione teorica di questo effetto nel caso non lineare parte da un'osservazione di base. lo Jacobiano della mappa di srotolamento $\partial z_K / \partial \mathbf{a}$ per un predittore non lineare f_θ è il prodotto di Jacobiani parziali

$$J_\Phi(\mathbf{a}) = [\partial f_\theta / \partial z]_{z_{K-1}} \cdot [\partial f_\theta / \partial z]_{z_{K-2}} \cdot \ldots \cdot [\partial f_\theta / \partial z]_{z_1} \cdot \partial f_\theta / \partial \mathbf{a} , \quad (7.16)$$

dove ogni fattore è lo Jacobiano locale di f_θ rispetto allo stato nel punto corrispondente della traiettoria. La struttura (7.16) è analoga alla struttura lineare $J_\Phi = [A^{K-1}B, \ldots, B]$, con le matrici A^k sostituite dai prodotti di Jacobiani locali. La rettitudine delle traiettorie, imponendo che le variazioni di stato consecutive siano quasi nella stessa direzione, riduce la variabilità di questi Jacobiani locali lungo la traiettoria: se lo stato evolve quasi linearmente, lo Jacobiano di f_θ varia poco da un passo al successivo, approssimando localmente la struttura lineare con A quasi costante. Questa interpretazione heuristica suggerisce che la garanzia formale (7.11) si estenda, in forma approssimata, al caso non lineare con dinamica a bassa variazione di Jacobiano — una condizione che il temporal straightening tende implicitamente a promuovere.

7.6 Connessioni con il controllo ottimale classico: il principio di Pontryagin e la forma hamiltoniana

L'analisi del condizionamento della perdita di pianificazione si arricchisce ulteriormente quando la si inquadra nell'ambito del principio del massimo di Pontryagin, che costituisce il fondamento variazionale della teoria del controllo ottimale per sistemi in tempo discreto con dinamica non lineare generale.

Per un sistema con dinamica $z_{t+1} = f(z_t, a_t)$ e costo terminale $\mathcal{L}(z_K) = \|z_K - z_g\|^2_2$, il principio di Pontryagin introduce le variabili aggiuntive $\lambda_t \in \mathbb{R}^d$ — denominate covariabili, o variabili aggiunte, o semplicemente moltiplicatori di Lagrange della dinamica vincolata — che evolvono secondo l'equazione retrograda

$$\lambda_t = [\partial f/\partial z(z_t, a_t)]^\top \lambda_{t+1} , \quad t = K-1, K-2, ..., 0 , \qquad (7.17)$$

con condizione terminale $\lambda_K = \nabla_{z_K} \mathcal{L}(z_K) = 2(z_K - z_g)$. La condizione di stazionarietà rispetto alle azioni stabilisce che, in corrispondenza di un ottimo, il gradiente dell'hamiltoniano rispetto alle azioni deve essere nullo:

$$\partial\mathcal{H}/\partial a_t = [\partial f/\partial a(z_t, a_t)]^\top \lambda_{t+1} = 0 , \quad t = 0, 1, ..., K-1 , \quad (7.18)$$

dove $\mathcal{H}(z_t, a_t, \lambda_{t+1}) := \lambda_{t+1}^\top f(z_t, a_t)$ è la funzione hamiltoniana. Le equazioni (7.17) e (7.18), insieme con la dinamica in avanti $z_{t+1} = f(z_t, a_t)$ e la condizione iniziale z_0, formano il sistema di condizioni di ottimalità del primo ordine: il sistema Hamiltoniano a due punti di confine, o BVP (Boundary Value Problem) hamiltoniano.

Nel caso lineare con $f(z_t, a_t) = Az_t + Ba_t$, il sistema si semplifica notevolmente. Le equazioni aggiuntive diventano $\lambda_t = A^\top \lambda_{t+1}$, con $\lambda_K = 2(z_K - z_g)$, da cui

$$\lambda_t = (A^\top)^{K-t} \lambda_K = 2(A^\top)^{K-t}(z_K - z_g) . \qquad (7.19)$$

La condizione di stazionarietà $\partial\mathcal{H}/\partial a_t = B^\top \lambda_{t+1} = 0$ per ogni t, sostituita con (7.19), fornisce

$$B^\top (A^\top)^{K-1-t} \cdot 2(z_K - z_g) = 0 \quad \text{per } t = 0, ..., K-1 , \qquad (7.20)$$

ossia $J_\Phi^\top (z_K - z_g) = 0$, che è esattamente la condizione del gradiente $\nabla_{\mathbf{a}} \mathcal{L}(\mathbf{a}) = 2\, J_\Phi^\top (z_K - z_g) = 0$. La coerenza del formalismo è dunque confermata: le condizioni di Pontryagin riducono alle condizioni standard del gradiente discendente nel caso lineare.

L'interesse di questa riformulazione non è puramente formale. Essa suggerisce una lettura del processo di pianificazione a gradiente come propagazione retrograda di segnali di errore attraverso la dinamica latente — che è

esattamente ciò che l'algoritmo di retropropagazione attraverso il tempo (BPTT, backpropagation through time) implementa nei modelli del mondo neurali. La stabilità numerica di questa propagazione retrograda dipende dalla struttura spettrale di A, o più precisamente dei Jacobiani locali nel caso non lineare: se i valori singolari di A sono vicini all'unità — condizione garantita dalla ε-rettitudine — la propagazione retrograda non produce né esplosione né scomparsa del gradiente, e il segnale di errore giunge efficacemente alle azioni nelle prime posizioni dell'orizzonte.

Questa connessione chiarisce perché la rettitudine delle traiettorie latenti sia benefica non solo per la geometria della superficie di perdita ma anche per la stabilità numerica del calcolo del gradiente nelle implementazioni pratiche. In un modello del mondo con orizzonte $K = 25$ — il valore usato negli esperimenti di Wang et al. (2026) — la propagazione del gradiente attraverso K passi di dinamica non lineare è soggetta a fenomeni di esplosione o scomparsa se i Jacobiani locali hanno valori singolari lontani dall'unità. Il temporal straightening, imponendo implicitamente che i Jacobiani locali rimangano vicini all'identità nel senso discusso nella sezione precedente, stabilizza questa propagazione e rende il processo di ottimizzazione più affidabile.

7.7 Il Gramiano di controllabilità e il discrimine di raggiungibilità

Si è osservato che il Gramiano $W_K = \Sigma_{k=0}^{K-1} A^k BB^\top (A^\top)^k$ misura la controllabilità del sistema lineare (7.1) nell'orizzonte K. Un aspetto cruciale di questa misura, spesso trascurato nella letteratura sull'apprendimento per la pianificazione, è il suo ruolo come indicatore di raggiungibilità: uno stato $z_g \in \mathbb{R}^d$ è raggiungibile da z_0 in K passi se e solo se $z_g - A^K z_0$ appartiene allo spazio immagine di J_Φ, ossia di $W_K^{1/2}$.

In altri termini, il condizionamento di W_K non riguarda solo la velocità di convergenza dell'ottimizzatore, ma determina anche quali stati obiettivo sono raggiungibili con bassa energia di controllo e quali richiedono invece azioni di grande ampiezza. Per un sistema con W_K ben condizionato, ogni direzione dello spazio latente è accessibile con una quantità comparabile di energia di controllo: il sistema è, in un senso preciso, isotropicamente controllabile. Per un sistema con W_K mal condizionato, alcune direzioni richiedono energie di controllo enormemente maggiori di altre, e il pianificatore tende a convergere

verso soluzioni che evitano le direzioni difficili anche quando il target si trova in quelle direzioni.

Questo fenomeno ha una corrispondenza diretta con l'osservazione sperimentale che la distanza euclidea nello spazio latente dei modelli non raddrizzati non riflette la distanza geodetica nell'ambiente: se alcune direzioni latenti corrispondono a stati molto lontani geodeticamente ma vicini euclideanamente (perché la curvatura ha compresso quelle direzioni), il pianificatore che usa la distanza euclidea come funzione di costo riceve un segnale fuorviante e converge a configurazioni subottimali.

Il temporal straightening agisce precisamente su questa patologia. Imponendo che la traiettoria latente sia localmente rettilinea, esso vincola l'encoder a produrre uno spazio in cui le variazioni euclidee corrispondono a variazioni temporali unitarie — ossia a passi di transizione unitari nell'ambiente. In termini di Gramiano, questo equivale ad avvicinare W_K alla proporzionalità con l'identità: W_K ≈ cI_d per qualche costante c > 0, che è esattamente la condizione di controllabilità isotropica. Non si tratta di una garanzia formale derivabile direttamente dalla regolarizzazione di cosine similarity, ma di una corrispondenza che emerge dall'analisi congiunta della struttura geometrica imposta dall'obiettivo e delle proprietà del Gramiano.

Più precisamente, si può formulare il seguente argomento. Se A ≈ I — condizione promossa dalla rettitudine ε-small — allora A^k ≈ I per ogni k, e il Gramiano si approssima come

$$W_K \approx \Sigma_{k=0}^{K-1} I \cdot BB^\top \cdot I = K \cdot BB^\top . \qquad (7.21)$$

Il condizionamento di W_K è quindi dominato dal condizionamento di $BB^\top$, ossia $\varkappa(W_K) \approx \varkappa(BB^\top) = \varkappa(B)^2$, che è una proprietà della matrice di ingresso B e non cresce con l'orizzonte K. Questo recupera il bound (7.11) al prim'ordine in ε: per ε = 0 il condizionamento del Gramiano non dipende dall'orizzonte, e l'intera dipendenza da K è catturata dal termine $e^{6\varepsilon K}$, che per ε piccolo è quasi costante per gli orizzonti pratici.

7.8 Il collo di bottiglia informazionale e la pianificazione efficiente

Il quadro teorico sviluppato nelle sezioni precedenti si riferisce alla geometria differenziale delle traiettorie nello spazio latente e alle sue implicazioni per il condizionamento dell'ottimizzazione. Vi è tuttavia un'altra prospettiva, concettualmente complementare e matematicamente distinta, attraverso cui è possibile interpretare la funzione del temporal straightening: quella del collo di bottiglia informazionale.

L'approccio del collo di bottiglia informazionale, introdotto da Tishby, Pereira e Bialek nel 1999 in forma teorica e successivamente elaborato in connessione con l'apprendimento profondo, affronta il problema della rappresentazione come un problema di compressione ottimale con vincolo di rilevanza. Dato uno spazio di ingresso X — nel nostro caso, lo spazio delle osservazioni Ω — e una variabile target Y — nel contesto della pianificazione, l'insieme degli stati o delle sequenze di stati futuri rilevanti — si cerca una rappresentazione compressa Z che massimizzi la mutua informazione I(Z; Y) preservata verso il target, minimizzando simultaneamente la mutua informazione I(Z; X) con l'ingresso. Il compromesso tra compressione e rilevanza è regolato da un parametro $\beta \geq 0$, e l'obiettivo variazionale assume la forma

$$\mathcal{F}_{\mathrm{IB}}(Z) := I(Z; Y) - \beta \cdot I(Z; X) , \qquad (7.22)$$

che deve essere massimizzato sulla classe degli encoder $p(z \mid x)$.

L'interpretazione (7.22) ha una struttura duale rispetto all'obiettivo di addestramento di un modello del mondo. Nella formulazione JEPA, l'obiettivo di previsione minimizza il residuo tra lo stato latente predetto $\hat{z}_{t+1}$ e lo stato target z_{t+1}: in termini di mutua informazione, questa perdita promuove un'alta mutua informazione $I(Z_{t+1}; \hat{Z}_{t+1})$, il che — per la sufficienza dello stato latente rispetto alla dinamica — è equivalente a preservare le informazioni necessarie per prevedere lo stato futuro, ossia a mantenere un'alta $I(Z_t; Z_{t+1} \mid A_t)$. La regolarizzazione di curvatura del temporal straightening aggiunge un termine che non compare esplicitamente nella formulazione (7.22) ma che ha una naturale interpretazione informazionale.

Per rendere questo argomento preciso, si considera la seguente riformulazione. La cosine similarity tra velocità latenti consecutive misura il grado di allineamento direzionale tra i passi successivi della traiettoria. Un'alta cosine similarity implica che la variazione di stato al passo t+1 è quasi parallela alla

variazione di stato al passo t: la dinamica è quasi uniforme. Dal punto di vista informazionale, una traiettoria retta trasporta lo stesso segnale direzionale di una traiettoria curva, ma lo fa in modo più compresso: la struttura a bassa curvatura è una struttura a bassa variazione dei descrittori latenti, il che significa che la quantità di informazione necessaria per specificare l'evoluzione della traiettoria — dato il punto di partenza e la direzione iniziale — è minore per una traiettoria retta che per una curvilinea.

Questa intuizione suggerisce che il temporal straightening implementi implicitamente una forma di compressione informazionale selettiva: esso sopprime dall'encoder le informazioni che producono variazioni di curvatura nelle traiettorie latenti — ossia le informazioni non predittive della direzione futura del moto — preservando quelle che governano la propulsione verso l'obiettivo. In termini dell'approccio IB, il termine di curvatura agisce come un regolarizzatore su I(Z; X) che penalizza specificatamente la mutua informazione tra l'osservazione e quelle sue caratteristiche che sono irrilevanti per la previsione della direzione delle transizioni future.

7.9 La struttura del collo di bottiglia per la pianificazione

Si può formalizzare questa intuizione nel quadro dello IB condizionato all'azione, che appare più naturale per il contesto della pianificazione. Sia Z_t l'embedding dell'osservazione al tempo t, Z_{t+1} l'embedding al tempo t+1 e A_t l'azione compiuta. L'obiettivo del modello del mondo è, in primo luogo, apprendere una rappresentazione Z tale che Z_{t+1} sia prevedibile da (Z_t, A_t) con bassa perdita: questa condizione è catturata dalla mutua informazione condizionata

$$I(Z_{t+1}; Z_t \mid A_t) , \tag{7.23}$$

che deve essere massimizzata. Nello stesso tempo, l'encoder deve scartare le informazioni dell'osservazione che non contribuiscono alla prevedibilità della dinamica, il che si traduce nel minimizzare

$$I(Z_t; O_t \mid Z_{t-1}, A_{t-1}) , \tag{7.24}$$

ossia la mutua informazione tra l'embedding corrente e l'osservazione corrente al netto di ciò che è già specificato dall'embedding e dall'azione precedenti. Il termine (7.24) misura la quantità di informazione «nuova» che l'encoder preleva dall'osservazione al netto della componente già predetta dalla dinamica: minimizzarlo equivale a richiedere che l'encoder ignori le caratteristiche dell'osservazione che non alterano la traiettoria dinamica.

L'obiettivo congiunto del modello del mondo con temporal straightening può quindi essere letto, in termini informazionali, come la massimizzazione di un funzionale del tipo

$$\mathcal{J}_{\mathrm{WM}}(Z) = I(Z_{t+1}; \hat{Z}_{t+1}) - \lambda \cdot \mathscr{C}_{\mathrm{curv}}(Z) , \qquad (7.25)$$

dove $\mathscr{C}_{\mathrm{curv}}(Z)$ è una misura di curvatura delle traiettorie latenti — nel caso concreto, l'opposto della cosine similarity media — e $\lambda \geq 0$ è il parametro di regolarizzazione. Il primo termine di (7.25) promuove la rilevanza predittiva; il secondo promuove la geometria favorevole alla pianificazione.

La similitudine formale con l'obiettivo IB (7.22) è istruttiva, anche se non identica. Nel caso IB, il secondo termine è una misura di compressione — la mutua informazione $I(Z; X)$ — che penalizza la quantità assoluta di informazione preservata dall'osservazione. Nel caso del temporal straightening, il secondo termine è una misura geometrica — la curvatura delle traiettorie — che non corrisponde direttamente a una quantità di informazione. Tuttavia, vi è un senso in cui i due termini sono correlati: una rappresentazione fortemente compressa tende a produrre traiettorie più regolari, poiché il numero limitato di gradi di libertà dello spazio latente forza la struttura dinamica a diventare più semplice. La connessione non è necessaria — si possono costruire rappresentazioni compresse ma altamente curve, così come rappresentazioni ad alta dimensione ma rette — ma è una tendenza osservabile nella pratica.

Un'analogia più stringente emerge considerando il bottleneck nel contesto della prevedibilità della direzione delle transizioni. Si definisca come «informazione di navigazione» la componente dell'embedding che porta informazione sulla direzione e ampiezza della prossima transizione, e come «informazione contestuale» la componente che riflette aspetti dell'osservazione irrilevanti per la dinamica prossima futura. Il temporal straightening, richiedendo che la direzione delle transizioni successive sia allineata, penalizza indirettamente la variabilità della direzione della navigazione: ogni cambiamento direzionale nella

traiettoria produce un costo di curvatura. Ne consegue che l'encoder, per minimizzare tale costo, è incentivato a costruire rappresentazioni in cui i cambiamenti direzionali siano minimali, il che si ottiene eliminando le informazioni contestuali che perturbano la direzione corrente di transizione senza migliorare la prevedibilità della destinazione finale.

Questa lettura suggerisce che il temporal straightening operi come un collo di bottiglia selettivo: non comprime la quantità totale di informazione preservata, ma ne seleziona la struttura, favorendo le informazioni con coerenza temporale direzionale rispetto a quelle ad alta variabilità angolare.

7.10 Il bound di distorsione informazionale e la distanza geodetica

La proprietà più immediatamente verificabile del temporal straightening — quella che ne giustifica l'utilità pratica prima ancora di approfondirne le fondamenta teoriche — è il miglioramento dell'allineamento tra la distanza euclidea nello spazio latente e la distanza geodetica nell'ambiente. Questa proprietà merita un'analisi quantitativa che riveli le condizioni sotto cui tale allineamento è garantito o almeno promosso.

Si consideri un ambiente con spazio degli stati $\mathcal{S}$ e metrica geodetica d_g: $\mathcal{S} \times \mathcal{S} \to \mathbb{R}_{\geq 0}$. Si denoti con E: $\mathcal{S} \to \mathbb{R}^d$ l'encoder e con $\|E(s) - E(s')\|_2$ la distanza euclidea nell'embedding. Si dice che E è una mappa a distorsione limitata se esiste una costante $\varkappa \geq 1$ tale che

$$(1/\varkappa)\, d_g(s, s') \leq \|E(s) - E(s')\|_2 \leq \varkappa \cdot d_g(s, s') \quad \text{per ogni } s, s' \in \mathcal{S}. \tag{7.26}$$

La condizione (7.26) è quella di una quasi-isometria: l'encoder preserva le distanze geodetiche a meno di un fattore moltiplicativo $\varkappa$ in entrambe le direzioni. Un encoder per cui $\varkappa = 1$ è un'isometria, ossia preserva le distanze esattamente.

Per un encoder che produce traiettorie esattamente rettilinee con velocità di norma costante c, la distanza euclidea tra lo stato iniziale z_0 e lo stato terminale z_T è esattamente cT, mentre la distanza geodetica d_g(s_0, s_T) è proporzionale al numero di passi di transizione T se ogni passo ha lunghezza unitaria nell'ambiente. L'allineamento è dunque esatto per traiettorie rette con

velocità uniforme. Per traiettorie curve, la distanza euclidea $\|z_T - z_0\|_2$ è strettamente inferiore a cT per la disuguaglianza triangolare, mentre la distanza geodetica può essere maggiore o minore di T a seconda della geometria dell'ambiente, creando uno scarto che cresce con la curvatura della traiettoria.

La regolarizzazione di curvatura del temporal straightening promuove un'alta cosine similarity media, che — come mostrato nella sezione 7.4 — corrisponde a velocità latenti quasi parallele consecutive. In tale regime, la traiettoria latente si approssima con una semiretta e la distanza euclidea tra stato iniziale e finale è prossima a cT. Tuttavia, una traiettoria retta nello spazio latente corrisponde a una traiettoria retta nello spazio degli stati dell'ambiente? Questa domanda non può avere una risposta affermativa in generale: la 'rettitudine' è una proprietà dello spazio latente indotta dall'encoder, non una proprietà intrinseca dell'ambiente. L'encoder apprende a raddrizzare le traiettorie attraverso l'obiettivo congiunto di previsione e curvatura, ma la corrispondenza tra rettitudine latente e rettitudine geodetica è una proprietà che emerge dall'addestramento e dipende dalla qualità dei dati.

L'evidenza sperimentale di Wang et al. (2026) mostra tuttavia che, nelle mappe di calore delle distanze euclidee nello spazio latente addestrato con temporal straightening, la struttura spaziale del labirinto — incluse le pareti e i passaggi — è riprodotta fedelmente dalla geometria degli embedding: le regioni del labirinto che richiedono molti passi per essere raggiunte corrispondono a embedding distanti, e viceversa. Questa corrispondenza si ottiene nonostante l'addestramento sia condotto su traiettorie subottimali non esperte, suggerendo che il temporal straightening incoraggia l'encoder a costruire una rappresentazione in cui la struttura causale della dinamica — quanti passi sono necessari per transitare tra due stati — si riflette direttamente nella metrica euclidea, indipendentemente dalla struttura visiva delle osservazioni.

7.11 La tensione tra anti-collasso, rettitudine e compressione

La monografia ha dedicato ampio spazio, nelle sezioni precedenti e nell'appendice C, all'analisi delle strategie anti-collasso per gli encoder in architetture JEPA. La sezione corrente introduce una dimensione supplementare di analisi: il temporal straightening impone un vincolo geometrico sulle traiettorie latenti che si sovrappone e interagisce con il vincolo anti-collasso.

La questione fondamentale è se questi due obiettivi — evitare il collasso degli embedding e promuovere la rettitudine delle traiettorie — siano strutturalmente compatibili o in tensione. La risposta emerge dall'esame delle soluzioni degeneri.

Una soluzione collassata z_t = costante per ogni t è trivialmente retta: una traiettoria costante ha curvatura nulla ovunque, poiché i vettori di velocità $v_t = z_{t+1} - z_t = 0$ non hanno direzione definita e la cosine similarity è priva di significato. Questo mostra che la minimizzazione della sola funzione di curvatura non è sufficiente a prevenire il collasso: la soluzione degenere è, in senso limite, un ottimo globale della regolarizzazione di rettitudine.

La compatibilità con l'obiettivo di previsione risolve questa degenerazione. L'obiettivo di previsione, che minimizza $\|\hat{z}_{t+1} - \text{sg}(z_{t+1})\|^2_2$, richiede che lo stato predetto dal predittore sia vicino allo stato effettivamente osservato. Se tutti gli embedding collassano alla stessa costante, lo stato predetto è uguale allo stato target — il che sembrerebbe soddisfare l'obiettivo di previsione con perdita nulla — ma soltanto perché l'operazione di stop-gradient nel target previene la retroazione che renderebbe il collasso un punto fisso stabile. La strategia di stop-gradient è dunque non solo una scelta implementativa di convenienza, ma un meccanismo strutturalmente necessario per evitare che l'obiettivo combinato sia dominato dalla soluzione degenere di rettitudine-senza-contenuto.

Formalmente, l'obiettivo totale $\mathcal{L}_{total} = \mathcal{L}_{pred} + \lambda\mathcal{L}_{curv}$ con stop-gradient nel ramo target può essere analizzato come un gioco a due agenti asimmetrico: il predittore f_θ tenta di prevedere z_{t+1} nel suo spazio di arrivo; l'encoder E_ϕ tenta di produrre traiettorie rette nel proprio spazio di partenza. La soluzione collassata è un punto fisso del solo agente-encoder (rettitudine massimale) ma non dell'agente-predittore (previsione non informativa). Questa struttura di gioco, analoga alla struttura dei metodi stop-gradient nella letteratura sulla rappresentazione autosupervisionata, garantisce che l'ottimizzazione congiunta converga verso soluzioni non collassate con alta probabilità, come confermato dalla stabilità delle procedure di addestramento riportata in Wang et al. (2026).

Un'ulteriore tensione riguarda il rapporto tra rettitudine e compressione della dimensionalità. Come osservato sperimentalmente, la riduzione da 384 a 8 canali attraverso un proiettore spaziale non degrada la performance di pianificazione, proprio perché la regolarizzazione di curvatura produce una

struttura quasi unidimensionale locale — le traiettorie sono quasi rettilinee, quindi ogni loro segmento è quasi contenuto in una semiretta — che è preservabile con pochi canali. In termini di IB, questo si traduce in una riduzione della dimensione effettiva del collo di bottiglia necessaria per raggiungere un dato livello di rilevanza predittiva: la rettitudine riduce la dimensione intrinseca dello spazio delle traiettorie, consentendo una compressione più aggressiva senza perdita di informazione pianificazione-rilevante.

7.12 Prospettive sull'estensione a dinamiche non lineari e ambienti ricchi

Le analisi svolte nelle sezioni precedenti hanno utilizzato la linearità della dinamica latente come ipotesi semplificatrice per derivare garanzie formali precise. Ci si chiede dunque in quale misura queste garanzie si estendano al caso di predittori non lineari, che costituisce la situazione effettiva degli esperimenti.

Per predittori non lineari, la perdita di pianificazione $\mathcal{L}(\mathbf{a})$ non è più quadratica in $\mathbf{a}$ e lo Hessiano non è più costante. Ciò che rimane è la struttura locale: per ogni punto $\mathbf{a}$ nello spazio delle azioni, la perdita è localmente approssimabile con una forma quadratica con Hessiano $H(\mathbf{a}) = \nabla^2_{\mathbf{a}} \mathcal{L}(\mathbf{a})$, e le proprietà di convergenza locale del gradiente discendente dipendono dalla struttura spettrale di $H(\mathbf{a})$.

L'analogo del bound (7.11) nel caso non lineare richiederebbe il controllo dei prodotti di Jacobiani locali $\partial f_\theta / \partial z$ lungo la traiettoria, come discusso nella sezione 7.5. La rettitudine delle traiettorie latenti, imponendo che le variazioni di stato successive siano quasi parallele, riduce la variabilità di questi Jacobiani locali: se la traiettoria evolve quasi uniformemente, il predittore opera sempre in regimi di input simili e la variazione del suo Jacobiano è limitata. Questa argomentazione è essenzialmente di natura locale e vale nell'intorno di traiettorie di bassa curvatura.

Per ambienti più ricchi e ad alta dimensione — ambiti robotici tridimensionali, manipolazione di oggetti complessi, navigazione in scenari non strutturati — la validità della linearizzazione locale si riduce e la dinamica può presentare cambiamenti di regime significativi. In questi contesti, il temporal straightening rimane uno strumento di regolarizzazione che promuove la coerenza

direzionale delle traiettorie, ma le garanzie teoriche derivate nel caso lineare diventano approssimazioni sempre più grossolane.

Una direzione di ricerca promettente è la connessione con la geometria differenziale delle varietà riemanniane apprese. Se si interpreta lo spazio latente come una varietà riemanniana dotata di una metrica indotta dall'encoder — la metrica pullback dalla distanza euclidea in $\mathbb{R}^d$ attraverso la mappa inversa dell'encoder — allora il temporal straightening equivale a richiedere che le traiettorie ammissibili dell'ambiente si mappino su geodetiche della varietà latente rispetto alla metrica euclidea. Questa è precisamente la proprietà che definisce una mappa armonica tra varietà riemanniane. Formulare il temporal straightening come approssimazione dell'armonicità dell'encoder aprirebbe la strada all'applicazione di tecniche dalla geometria differenziale computazionale — calcolo della curvatura riemanniana dello spazio latente appreso e sua minimizzazione attraverso metodi variazionali — che potrebbero produrre garanzie più forti di quelle attualmente disponibili nel quadro lineare.

7.13 Sintesi: una triplice equivalenza concettuale

Le analisi condotte nelle sezioni precedenti convergono verso una visione unitaria che può essere sintetizzata come una triplice equivalenza concettuale — non formalmente identica ma strutturalmente coerente — tra tre famiglie di proprietà dello spazio latente.

La prima famiglia è quella geometrica: un encoder produce uno spazio latente favorevole alla pianificazione quando le traiettorie ammissibili in tale spazio hanno bassa curvatura, ossia quando i vettori di velocità latente successivi sono quasi paralleli. Questa proprietà è misurabile attraverso la cosine similarity e ottimizzabile attraverso la regolarizzazione corrispondente.

La seconda famiglia è quella algebrico-dinamica: uno spazio latente con traiettorie rettilinee corrisponde, nella dinamica lineare approssimante, a una matrice di transizione A vicina all'identità. Questa proprietà implica che il Gramiano di controllabilità W_K sia ben condizionato e quasi proporzionale all'identità, garantendo che lo Hessiano della perdita di pianificazione abbia condizionamento limitato e non crescente rapidamente con l'orizzonte K. Un Hessiano ben condizionato si traduce in una convergenza rapida del gradiente discendente e in un paesaggio di perdita quasi-convesso.

La terza famiglia è quella informazionale: un encoder con traiettorie latenti rettilinee implementa un collo di bottiglia selettivo che comprime le informazioni ad alta variabilità angolare — irrilevanti per la prevedibilità della direzione delle transizioni future — preservando quelle a coerenza direzionale temporale. Questa compressione selettiva è la controparte informazionale della rettitudine geometrica e si manifesta praticamente nella possibilità di comprimere aggressivamente la dimensionalità delle feature senza perdita di performance nella pianificazione.

Queste tre famiglie non sono descrizioni indipendenti dello stesso fenomeno: sono proiezioni di un'unica struttura su tre spazi concettuali distinti. La struttura comune è quella di uno spazio di rappresentazione in cui l'encoder ha appreso a organizzare gli stati dell'ambiente secondo un ordinamento temporale coerente, nel quale la distanza euclidea riflette la distanza causale — il numero minimo di transizioni necessarie per passare da uno stato a un altro — piuttosto che la dissimilarità visiva o strutturale. Questa nozione di distanza causale è il nucleo concettuale di ciò che i modelli del mondo cercano di apprendere, e il temporal straightening è, nella lettura qui sviluppata, lo strumento che costringe l'encoder a privilegiare sistematicamente questa metrica su qualsiasi altra.

La connessione con la teoria del controllo ottimale e con il formalismo del collo di bottiglia informazionale non è accessoria: essa rivela che il temporal straightening non è un'euristica motivata pragmaticamente, bensì un principio che trova fondamento in due tradizioni teoriche consolidate e indipendenti. Che la rettitudine delle traiettorie latenti migliori il condizionamento del Gramiano di controllabilità è una conseguenza della teoria lineare dei sistemi; che essa promuova la compressione delle informazioni pianificazione-irrilevanti è una conseguenza dei principi del collo di bottiglia informazionale. L'originalità del contributo di Wang et al. (2026) sta nell'aver riconosciuto che entrambe le proprietà emergono dalla stessa operazione di regolarizzazione e si traducono, empiricamente, in miglioramenti sostanziali e sistematici della performance di pianificazione su una classe diversificata di ambienti.

7.14 Implicazioni per la progettazione di sistemi di pianificazione latente

Le considerazioni teoriche sviluppate nelle sezioni precedenti suggeriscono alcune implicazioni di ordine progettuale per la costruzione di sistemi di pianificazione basati su modelli del mondo latenti.

La prima riguarda la scelta dell'encoder. Un backbone con elevata flessibilità rappresentazionale — come un ResNet addestrato da zero — ha la capacità di adattare la propria geometria latente agli specifici requisiti di rettitudine imposti dalla regolarizzazione, producendo traiettorie intrinsecamente più rettilinee rispetto a un encoder basato su un backbone congelato cui si sovrappone solo un proiettore leggero. Tuttavia, la flessibilità ha un costo: richiede più dati per convergere e può produrre traiettorie di curvatura abnormalmente alta in assenza di regolarizzazione esplicita, come documentato dall'anomalia delle basse performance di ResNet senza straightening sugli ambienti Wall e PointMaze. Il temporal straightening trasforma questa fragilità in un vantaggio: con regolarizzazione esplicita, il ResNet produce sistematicamente le rappresentazioni più rette e le performance di pianificazione più elevate.

La seconda implicazione riguarda la dimensionalità delle feature. Poiché la rettitudine riduce la dimensione intrinseca dello spazio delle traiettorie latenti, la compressione dimensionale può essere spinta più aggressivamente in presenza di raddrizzamento. La riduzione da 384 a 8 canali documentata preservando la risoluzione spaziale dei patch token — non degrada la performance proprio perché la regolarizzazione di curvatura produce una struttura quasi unidimensionale locale preservabile con pochi canali. Questa osservazione ha conseguenze dirette sull'efficienza computazionale dei sistemi di pianificazione: spazi latenti più compatti riducono il costo di ogni rollout del modello del mondo e accelerano l'ottimizzazione delle azioni.

La terza implicazione riguarda la scelta del metodo di pianificazione. L'analisi del condizionamento dell'Hessiano mostra che, con temporal straightening, il paesaggio di perdita è sufficientemente regolare da rendere competitivo il gradiente discendente rispetto a metodi di ricerca stocastica computazionalmente onerosi come CEM o MPPI. La pianificazione a gradiente ha latenza molto inferiore e scala meglio con la lunghezza dell'orizzonte, poiché il suo costo per iterazione è O(K) — un forward pass e un backward pass di lunghezza K — contro O(N·K) per CEM, con N tipicamente dell'ordine di 200 campioni per performance competitive. La riduzione del divario tra GD e CEM documentata negli esperimenti — riduzione che raggiunge l'azzeramento completo su alcuni ambienti — quantifica il vantaggio computazionale reso accessibile dal raddrizzamento.

La quarta implicazione, più speculativa, riguarda la connessione con la teoria dell'apprendimento per rinforzo. Il temporal straightening può essere interpretato come un meccanismo che promuove la struttura di funzione di valore nelle rappresentazioni latenti: se la distanza euclidea nello spazio latente si allinea con la distanza geodetica nell'ambiente — proporzionale al costo minimo di raggiungimento dello stato target — allora lo spazio latente risultante ha una struttura analoga a quella della funzione di valore ottimale $V^*(s)$. Questa connessione non è sviluppata formalmente nel lavoro originale, ma rappresenta una direzione teorica feconda: l'analisi del temporal straightening come metodo per apprendere implicitamente una funzione di valore geometrica nello spazio latente aprirebbe la strada all'uso di tecniche dell'apprendimento per rinforzo — equazione di Bellman, temporal difference learning — per affinare ulteriormente la geometria dell'encoder in direzione di una coerenza causale ancora più stringente.

7.15 Sintesi: il dialogo teorico tra LeWorldModel e il raddrizzamento esplicito

Le analisi condotte in questo capitolo consentono ora di mettere in dialogo esplicito i due lavori attorno ai quali la monografia si è strutturata, e di identificare la congiunzione teoricamente più ricca tra di essi. Maes et al. (2026) dimostrano che il raddrizzamento temporale emerge spontaneamente come proprietà indotta dall'obiettivo JEPA combinato con la regolarizzazione SIGReg: l'encoder non è mai istruito a produrre traiettorie rettilinee, eppure le produce, con un valore di cosine similarity media che supera quello di PLDM nonostante quest'ultimo ottimizzi esplicitamente un termine di smoothness temporale. Questo raddrizzamento implicito è interpretato nella sezione 7.3 come la conseguenza del fatto che la prevedibilità delle transizioni e la gaussianità degli embedding impongono congiuntamente un vincolo geometrico sulle traiettorie — un vincolo che favorisce la rettitudine per ragioni informazionali prima ancora che metriche.

Wang et al. (2026) partono da questa proprietà emergente e la amplificano: aggiungendo un termine esplicito di regolarizzazione della curvatura all'obiettivo di addestramento, si chiede all'encoder di massimizzare attivamente la cosine similarity tra velocità latenti consecutive, anziché lasciarle aumentare come effetto collaterale non intenzionale. Le conseguenze di questa amplificazione, documentate nella Tabella 3 del loro paper, sono misurabili e sostanziali: il gap tra gradiente discendente e CEM — il gap che nella versione

originale di LeWorldModel aveva motivato la scelta del CEM come pianificatore predefinito — si azzera su ambienti come UMaze e PointMaze. In altri termini, Wang et al. rimuovono sperimentalmente uno dei limiti strutturali che LeWorldModel aveva accettato come inevitabile.

La congiunzione teorica tra i due lavori può essere enunciata con precisione. LeWorldModel dimostra che il raddrizzamento è un fenomeno emergente dell'architettura JEPA: esso non ha bisogno di essere imposto dall'esterno perché l'obiettivo combinato lo genera come soluzione a bassa curvatura del problema di ottimizzazione. Wang et al. dimostrano che, una volta amplificato esplicitamente, questo stesso raddrizzamento riduce il condizionamento del Gramiano di controllabilità — in accordo con la disuguaglianza (7.11) — a tal punto da rendere competitivo il gradiente discendente con il CEM. Il filo teorico che collega i due risultati è la stima (7.11) stessa: LeWorldModel produce un ε naturalmente piccolo, Wang et al. lo portano sistematicamente verso zero, e la disuguaglianza prevede correttamente l'impatto di questa riduzione sul condizionamento del pianificatore. Non si tratta di due lavori che si affiancano su tematiche parallele, ma di due tappe successive di uno stesso programma di ricerca sulla geometria degli spazi latenti per la pianificazione.

Da un punto di vista progettuale, questa lettura suggerisce una gerarchia di strategie. Un sistema che utilizzi LeWorldModel senza raddrizzamento esplicito ottiene un ε spontaneamente ridotto ma non minimizzato, e deve affidarsi al CEM per compensare la curvatura residua del paesaggio di perdita — un costo computazionale di ordine $O(N \cdot K)$ a ogni ciclo di pianificazione, con N tipicamente pari a 200 campioni. Un sistema che integri il raddrizzamento esplicito di Wang et al. all'interno dell'architettura LeWorldModel — aggiungendo il termine di regolarizzazione della curvatura all'obiettivo SIGReg — ottiene un ε prossimo a zero e può pianificare con il gradiente discendente a costo $O(K)$, mantenendo al contempo le garanzie anti-collasso distribuzionali di SIGReg che il solo raddrizzamento non fornisce. Questa integrazione non è ancora stata realizzata nella letteratura al momento della stesura della monografia, e costituisce una direzione naturale di sviluppo.

Appendice al Capitolo 7 — Dimostrazioni complete e questioni aperte

A7.1 Dimostrazione completa del Lemma sul Hessiano e l'equivalenza con il Gramiano

L'identificazione tra il condizionamento del Hessiano di pianificazione e il condizionamento del Gramiano di controllabilità — enunciata nella sezione 7.2 come identità algebrica — merita di essere percorsa passo per passo, poiché ogni passaggio rivela una proprietà strutturale del problema che non è immediatamente ovvia dalla semplice ispezione del risultato finale. Si riportano prima gli enunciati formali — Definizioni, Ipotesi, Lemma — così come compaiono nell'Appendice C di Wang et al. (2026), cui seguono le dimostrazioni estese.

Ipotesi C.1 — Dinamica latente lineare (Wang et al., 2026). Si assume una dinamica latente lineare della forma z_{t+1} = Az_t + Ba_t, con A ∈ ℝ^{d×d} e B ∈ ℝ^{d×d_a}.

Definizione C.2 — Numero di condizionamento effettivo (Wang et al., 2026). Per una matrice semidefinita positiva H ≽ 0 con nucleo non banale, si definisce il numero di condizionamento effettivo come $\varkappa_{\mathrm{eff}}(H) := \sigma_{\max}(H) / \sigma^{+}_{\min}(H)$, dove $\sigma^{+}_{\min}(H)$ è il più piccolo valore singolare non nullo di H.

Definizione C.3 — Transizione ε-retta (Wang et al., 2026). Sotto la dinamica lineare dell'Ipotesi C.1, si definisce $\varepsilon := \|A - I\|_2$. Il termine «retta» riflette che, al tendere di ε a 0, la dinamica di f si avvicina a quella della funzione di riferimento g : (z_t, a_t) ↦ z_t + Ba_t, in cui lo stato evolve linearmente lungo una traiettoria rettilinea modificata solo dall'ingresso di controllo.

Lemma C.4 — Forma del Hessiano ed equivalenza con il Gramiano (Wang et al., 2026). Sotto le equazioni (14)–(15) di Wang et al. (2026), lo Hessiano di pianificazione soddisfa $H := \nabla^2_a \mathcal{L}(a) = 2J_\Phi^\top J_\Phi \succeq 0$. Inoltre, i valori singolari non nulli di $J_\Phi^\top J_\Phi$ coincidono con quelli di $J_\Phi J_\Phi^\top$, cosicché $\varkappa_{\mathrm{eff}}(H) = \varkappa(W_K)$. □ (La dimostrazione dettagliata segue.)

Il punto di partenza è l'espressione della perdita terminale nella pianificazione a orizzonte K per un sistema lineare:

$$\mathcal{L}(**a**) = \|z_K - z_g\|_2^2 ,$$

con $z_K = A^K z_0 + \Sigma_{t=0}^{K-1} A^{K-1-t} B a_t$.

Poiché z_0 e z_g sono fissati al momento dell'ottimizzazione su **a**, si può scrivere $z_K - z_g = J_\Phi \mathbf{a} + r$, dove $r := A^K z_0 - z_g$ è un vettore costante rispetto alle azioni e J_Φ è lo Jacobiano della mappa di rollout rispetto all'intera sequenza di azioni, organizzata come il vettore colonna $\mathbf{a} = (a_0^\top, a_1^\top, ..., a_{K-1}^\top)^\top \in \mathbb{R}^{Kd_a}$. lo Jacobiano ha la struttura a blocchi

$$J_\Phi = [A^{K-1}B,\ A^{K-2}B,\ ...,\ AB,\ B] \in \mathbb{R}^{d \times Kd_a} ,$$

dove il blocco in posizione (1, t+1) è $A^{K-1-t}B$ per $t = 0, 1, ..., K-1$. La perdita diventa allora

$$\mathcal{L}(\mathbf{a}) = \|J_\Phi \mathbf{a} + r\|_2^2 = \mathbf{a}^\top J_\Phi^\top J_\Phi \mathbf{a} + 2r^\top J_\Phi \mathbf{a} + \|r\|_2^2 .$$

Questa è una forma quadratica in **a** con matrice dei coefficienti del termine di secondo grado pari a $J_\Phi^\top J_\Phi$, che è semidefinita positiva per costruzione (essendo il prodotto di una matrice per la sua trasposta). lo Hessiano è dunque

$$H = \nabla_{\mathbf{a}}^2 \mathcal{L}(\mathbf{a}) = 2 J_\Phi^\top J_\Phi \succeq 0 .$$

Fin qui, nulla di sorprendente. L'identificazione cruciale con il Gramiano emerge dal confronto tra H e W_K. Per definizione del Gramiano,

$$W_K = J_\Phi J_\Phi^\top = \Sigma_{k=0}^{K-1} A^k B B^\top (A^\top)^k \in \mathbb{R}^{d \times d} ,$$

e $H/2 = J_\Phi^\top J_\Phi \in \mathbb{R}^{Kd_a \times Kd_a}$. Questi due oggetti hanno dimensioni diverse — W_K è d×d mentre $J_\Phi^\top J_\Phi$ è $Kd_a \times Kd_a$ — ma condividono i valori singolari non nulli. Questo fatto, standard nell'algebra lineare, segue dall'osservazione che per qualsiasi matrice M, i valori singolari di $MM^\top$ e $M^\top M$ sono identici (entrambi sono i quadrati dei valori singolari di

M). Applicando questo con M = J_Φ si ottiene che i valori singolari non nulli di W_K = J_Φ J_Φ^⊤ coincidono con quelli di H/2 = J_Φ^⊤ J_Φ. Poiché il numero di condizionamento effettivo è il rapporto tra il valore singolare massimo e quello minimo non nullo, si ha

$$\varkappa_{\text{eff}}(H) = \sigma_{\max}(H) / \sigma^{+}_{\min}(H) = \sigma_{\max}(2J_\Phi^\top J_\Phi) / \sigma^{+}_{\min}(2J_\Phi^\top J_\Phi)$$

$$= \sigma_{\max}(W_K) / \sigma^{+}_{\min}(W_K) = \varkappa(W_K) .$$

L'uguaglianza è esatta, non un'approssimazione: il condizionamento del problema di ottimizzazione della pianificazione è identicamente il condizionamento del Gramiano di controllabilità. Questo stabilisce un ponte formale — non meramente analogico — tra la teoria dell'ottimizzazione e la teoria del controllo.

A7.2 Dimostrazione completa del Teorema sul condizionamento del Gramiano

Il Teorema 4.4 del paper di Wang et al. (2026) deriva il bound sul condizionamento attraverso una stima separata dall'alto di $\sigma_{\max}(W_K)$ e dal basso di $\sigma_{\min}(W_K)$, valida nel caso $d_a = d$ con B invertibile. Si riporta prima l'enunciato formale completo, poi si ripercorre la dimostrazione arricchendola di commenti che ne chiariscono le ipotesi.

Teorema C.5 — Bound sul condizionamento (Wang et al., 2026; corrisponde al Teorema 4.4 del paper). Sotto l'Ipotesi C.1, si consideri prima il caso $d_a = d$ con B invertibile. Allora

$$\varkappa_{\text{eff}}(H) = \varkappa(W_K) \le \varkappa(B)^2 \cdot [\Sigma_{k=0}^{K-1} \sigma_{\max}(A)^{2k}] / [\Sigma_{k=0}^{K-1} \sigma_{\min}(A)^{2k}] \le \varkappa(B)^2 \varkappa(A)^{2(K-1)} , \qquad (21)$$

dove $\varkappa(A) := \sigma_{\max}(A)/\sigma_{\min}(A)$. Inoltre, se la transizione è ε-retta con $\varepsilon = \|A - I\|_2 < 1$, allora

$$\varkappa_{\text{eff}}(H) \le \varkappa(B)^2 \cdot ((1 + \varepsilon)/(1 - \varepsilon))^{2(K-1)} \le \varkappa(B)^2 e^{6\varepsilon K} \quad \text{per } \varepsilon \le 1/2 . \qquad (22)$$

Dimostrazione. Per il Lemma C.4 è sufficiente stimare $\varkappa(W_K)$; si veda la sezione seguente. □

Osservazione C.6 — Azioni a bassa dimensionalità (Wang et al., 2026). Quando $d_a < d$, la matrice B non è invertibile e W_K (e quindi H) può essere singolare al di fuori del sottospazio controllabile. Il Teorema C.5 vale sul sottospazio $S_K = \mathrm{range}(W_K)$ calcolando $\varkappa_{\mathrm{eff}}$ mediante $\sigma^{+}_{\min}(W_K)$; in tal caso sono necessarie ulteriori ipotesi di controllabilità per stimare dal basso $\sigma^{+}_{\min}(W_K)$.

Stima dall'alto di $\lambda_{\max}(W_K)$. Per qualsiasi vettore unitario $x \in \mathbb{R}^d$, il valore della forma bilineare $x^\top W_K x$ si scrive come

$$x^\top W_K x = \sum_{k=0}^{K-1} x^\top A^k B B^\top (A^\top)^k x = \sum_{k=0}^{K-1} \| B^\top (A^\top)^k x \|_2^2 .$$

Per la submoltiplicatività della norma spettrale, $\| B^\top (A^\top)^k x \|_2 \le \| B^\top \|_2 \cdot \| (A^\top)^k \|_2 \cdot \|x\|_2 = \sigma_{\max}(B) \cdot \sigma_{\max}(A)^k \cdot 1 = \sigma_{\max}(B) \cdot \sigma_{\max}(A)^k$. Di conseguenza

$$x^\top W_K x \le \sigma_{\max}(B)^2 \cdot \sum_{k=0}^{K-1} \sigma_{\max}(A)^{2k} ,$$

e il massimante su $\|x\|_2 = 1$ fornisce

$$\lambda_{\max}(W_K) \le \sigma_{\max}(B)^2 \cdot \sum_{k=0}^{K-1} \sigma_{\max}(A)^{2k} . \quad (A7.1)$$

Stima dal basso di $\lambda_{\min}(W_K)$. Qui interviene in modo essenziale l'ipotesi che B sia invertibile, ossia $d_a = d$ e $\sigma_{\min}(B) > 0$. Per ogni vettore unitario $x \in \mathbb{R}^d$ e per ogni $k \ge 0$ si ha

$$\| B^\top (A^\top)^k x \|_2 \ge \sigma_{\min}(B^\top) \cdot \| (A^\top)^k x \|_2 = \sigma_{\min}(B) \cdot \| (A^\top)^k x \|_2 .$$

Poiché $(A^\top)^k$ è una matrice invertibile (per invertibilità di A, anch'essa implicata dall'ipotesi), si ha $\|(A^\top)^k\, x\|_2 \geq \sigma_{\min}(A^\top)^k \cdot \|x\|_2 = \sigma_{\min}(A)^k$. Dunque

$$x^\top W_K\, x \geq \sigma_{\min}(B)^2 \cdot \sum_{k=0}^{K-1} \sigma_{\min}(A)^{2k} \,,$$

e il minimo su $\|x\|_2 = 1$ fornisce

$$\lambda_{\min}(W_K) \geq \sigma_{\min}(B)^2 \cdot \sum_{k=0}^{K-1} \sigma_{\min}(A)^{2k} \,. \tag{A7.2}$$

Combinazione. Dividendo (A7.1) per (A7.2) si ottiene il primo bound del Teorema 4.4:

$$\kappa(W_K) \leq \kappa(B)^2 \cdot \left[\sum_{k=0}^{K-1} \sigma_{\max}(A)^{2k}\right] / \left[\sum_{k=0}^{K-1} \sigma_{\min}(A)^{2k}\right] . \tag{A7.3}$$

Per derivare il secondo bound $\kappa(W_K) \leq \kappa(B)^2\, \kappa(A)^{2(K-1)}$, si usa la stima grezza

$$\left[\sum_{k=0}^{K-1} \sigma_{\max}(A)^{2k}\right] / \left[\sum_{k=0}^{K-1} \sigma_{\min}(A)^{2k}\right]$$

$$\leq \max_{0 \leq k \leq K-1} \left[\sigma_{\max}(A)^{2k} / \sigma_{\min}(A)^{2k}\right]$$

$$= \kappa(A)^{2(K-1)} \,,$$

dove l'ultimo passaggio usa il fatto che il massimo del rapporto è raggiunto a $k = K-1$ (essendo $\kappa(A) \geq 1$ per definizione). Questo produce

$$\kappa(W_K) \leq \kappa(B)^2 \cdot \kappa(A)^{2(K-1)} \,. \tag{A7.4}$$

Specializzazione al caso ε-retto. Se la transizione è ε-retta con $\varepsilon = \|A - I\|_2 < 1$, il lemma di perturbazione di Weyl stabilisce che $\sigma_{max}(A) \le 1 + \varepsilon$ e $\sigma_{min}(A) \ge 1 - \varepsilon$, da cui $\kappa(A) \le (1+\varepsilon)/(1-\varepsilon)$. Sostituendo in (A7.4) si ha

$$\kappa(W_K) \le \kappa(B)^2 \cdot ((1+\varepsilon)/(1-\varepsilon))^{2(K-1)} . \qquad (A7.5)$$

Per ottenere la forma esponenziale nel caso $\varepsilon \le 1/2$, si usa la disuguaglianza $\ln((1+\varepsilon)/(1-\varepsilon)) = \ln(1+\varepsilon) - \ln(1-\varepsilon) \le \varepsilon/(1-\varepsilon) + \varepsilon/(1-\varepsilon) \le 2\varepsilon/(1-\varepsilon)$. Per $\varepsilon \le 1/2$, $1/(1-\varepsilon) \le 2$, dunque $\ln((1+\varepsilon)/(1-\varepsilon)) \le 4\varepsilon$. Un'analisi leggermente più accurata, usando la serie di Taylor del logaritmo, dà $\ln((1+\varepsilon)/(1-\varepsilon)) = 2(\varepsilon + \varepsilon^3/3 + \varepsilon^5/5 + ...) \le 2\varepsilon/(1-\varepsilon^2) \le 3\varepsilon$ per $\varepsilon \le 1/2$. Quindi $((1+\varepsilon)/(1-\varepsilon)) \le e^{3\varepsilon}$, e

$$((1+\varepsilon)/(1-\varepsilon))^{2(K-1)} \le e^{6\varepsilon(K-1)} \le e^{6\varepsilon K} ,$$

da cui il bound finale del paper:

$$\kappa_{eff}(H) = \kappa(W_K) \le \kappa(B)^2 \cdot e^{6\varepsilon K} . \qquad (A7.6)$$

Il valore di questo risultato nella prospettiva della pianificazione latente è precisamente quello discusso nella sezione 7.3: il bound (A7.6) mostra che il condizionamento cresce al più come $e^{6\varepsilon K}$, e non come e^{cK} per qualche costante c fissa, il che significa che per ε sufficientemente piccolo l'orizzonte K può essere esteso senza degrado dell'ottimizzazione. Per $\varepsilon = 0$ il condizionamento è $\kappa(B)^2$ indipendentemente dall'orizzonte.

A7.3 Dimostrazione completa della Proposizione sul proxy a cosine similarity

La Proposizione C.9 del paper è la stima fondamentale che collega la grandezza ottimizzata durante l'addestramento — la cosine similarity tra velocità latenti consecutive — con il parametro ε di rettitudine della dinamica. Si riportano prima gli enunciati formali completi, poi la dimostrazione estesa.

Ipotesi C.7 — Velocità latenti di norma costante e azioni a variazione limitata (Wang et al., 2026). Si definiscano le velocità latenti $v_t := z_{t+1} - z_t$. Si assume che esista una costante $c > 0$ tale che $\|v_t\|_2 = c$ per ogni $t = 0, ..., K - 1$, e che le azioni soddisfino la condizione di variazione limitata $\Delta a := \max_t \|a_{t+1} - a_t\|_2 < \infty$.

Definizione C.8 — Cosine similarity tra velocità latenti (Wang et al., 2026). Per $t = 0, ..., K - 2$, si definisce la cosine similarity al passo t come $\mathscr{C}_t := v_t^\top v_{t+1} / (\|v_t\|_2 \|v_{t+1}\|_2)$, e la cosine similarity media come $\mathscr{C} := (K - 1)^{-1} \Sigma_{t=0}^{K-2} \mathscr{C}_t$.

Proposizione C.9 — La cosine similarity come proxy di (A − I) lungo le direzioni visitate (Wang et al., 2026). Sotto la dinamica lineare dell'Ipotesi C.1, siano $\hat{v}_t := v_t/\|v_t\|_2$ i versori di velocità. Sotto l'Ipotesi C.7, per ogni $t = 0, ..., K - 2$:

$$\|(A - I)\,\hat{v}_t\|_2 \le \sqrt{2(1 - \mathscr{C}_t)} + \sigma_{\max}(B)\,\Delta a / c\,. \tag{23}$$

Se $\mathscr{C} \ge 1 - \eta$, allora

$$(K - 1)^{-1} \Sigma_{t=0}^{K-2} \|(A - I)\,\hat{v}_t\|_2 \le \sqrt{2\eta} + \sigma_{\max}(B)\,\Delta a / c\,. \tag{24}$$

Dimostrazione. Si veda la sezione seguente per la derivazione passo per passo. □

Formulazione precisa. Si assumano velocità latenti di norma costante $\|v_t\|_2 = c > 0$ per $t = 0, ..., K-1$, e variazione delle azioni limitata $\Delta a := \max_t \|a_{t+1} - a_t\|_2 < \infty$. Si definiscano la cosine similarity al passo t come $\mathscr{C}_t := v_t^\top v_{t+1} / (\|v_t\|_2 \|v_{t+1}\|_2)$ e la media su tutti i passi come $\mathscr{C} := (K-1)^{-1} \Sigma_{t=0}^{K-2} \mathscr{C}_t$.

Derivazione dell'equazione chiave. Sotto la dinamica lineare $z_{t+1} = Az_t + Ba_t$, la velocità $v_{t+1} = z_{t+2} - z_{t+1}$ si scrive come

$$\begin{aligned} v_{t+1} &= Az_{t+1} + Ba_{t+1} - Az_t - Ba_t \\ &= A(z_{t+1} - z_t) + B(a_{t+1} - a_t) \\ &= Av_t + B(a_{t+1} - a_t)\,. \end{aligned}$$

Dunque $v_{t+1} - v_t = (A - I)v_t + B(a_{t+1} - a_t)$, e dividendo per $c = \|v_t\|_2$

$$v_{t+1}/c - \hat{v}_t = (A - I)\hat{v}_t + B(a_{t+1} - a_t)/c \,,$$

da cui per la disuguaglianza triangolare

$$\|(A - I)\hat{v}_t\|_2 \le \|v_{t+1}/c - \hat{v}_t\|_2 + \|B(a_{t+1} - a_t)\|_2/c$$

$$\le \|v_{t+1}/c - \hat{v}_t\|_2 + \sigma_{\max}(B)\Delta a/c \,. \qquad \text{(A7.7)}$$

Il primo termine di (A7.7) è la distanza euclidea tra il versore $\hat{v}_{t+1} := v_{t+1}/\|v_{t+1}\|_2$ scalato all'unità e $\hat{v}_t$, il che — per l'identità $\|u - w\|^2_2 = 2 - 2u^\top w$ per vettori unitari — non è esattamente quello che compare. Ciò che compare è $\|v_{t+1}/c - \hat{v}_t\|_2$, che dipende sia dalla direzione che dalla norma di v_{t+1}. Poiché $\|v_{t+1}\|_2 = c$ per ipotesi, v_{t+1}/c è anch'esso un versore uguale a $\hat{v}_{t+1}$, dunque $\|v_{t+1}/c - \hat{v}_t\|_2 = \|\hat{v}_{t+1} - \hat{v}_t\|_2$. Ora si usa l'identità geometrica

$$\|\hat{v}_{t+1} - \hat{v}_t\|^2_2 = \|\hat{v}_{t+1}\|^2_2 + \|\hat{v}_t\|^2_2 - 2\,\hat{v}_{t+1}^\top \hat{v}_t = 2(1 - \mathscr{C}_t) \,,$$

da cui $\|\hat{v}_{t+1} - \hat{v}_t\|_2 = \sqrt{(2(1 - \mathscr{C}_t))}$. Sostituendo in (A7.7)

$$\|(A - I)\hat{v}_t\|_2 \le \sqrt{(2(1 - \mathscr{C}_t))} + \sigma_{\max}(B)\Delta a/c \,. \qquad \text{(A7.8)}$$

La (A7.8) è la stima puntuale per il passo t. Mediando su $t = 0, \ldots, K-2$ e applicando la disuguaglianza di Jensen alla funzione concava $\sqrt{(\cdot)}$:

$$\frac{1}{K-1} \sum_t \|(A-I)\hat{v}_t\|_2$$

$$\le \frac{1}{K-1} \sum_t \sqrt{(2(1 - \mathscr{C}_t))} + \sigma_{\max}(B)\Delta a/c$$

$$\le \sqrt{\left(2 \cdot \frac{1}{K-1} \sum_t (1 - \mathscr{C}_t)\right)} + \sigma_{\max}(B)\Delta a/c$$

$$= \sqrt{(2(1 - \mathscr{C}))} + \sigma_{\max}(B)\Delta a/c \,.$$

Se $\mathscr{C} \geq 1 - \eta$, allora $\sqrt{(2(1 - \mathscr{C}))} \leq \sqrt{(2\eta)}$, e il bound diventa

$$(K-1)^{-1} \sum_{t=0}^{K-2} \|(A-I)\hat{v}_t\|_2 \leq \sqrt{(2\eta)} + \sigma_{max}(B)\Delta a/c \ . \quad (A7.9)$$

La (A7.9) è il risultato fondamentale: massimizzare la cosine similarity media — il che equivale a minimizzare η — riduce il membro sinistro di (A7.9), ossia la deviazione media di A dall'identità lungo le direzioni visitate dalla traiettoria.

Interpretazione e portata. Vale la pena soffermarsi su ciò che la (A7.9) afferma e su ciò che non afferma. Essa controlla $\|(A-I)\hat{v}_t\|_2$ per i versori $\hat{v}_t$ corrispondenti alle direzioni effettivamente percorse dalla traiettoria durante l'addestramento. Si tratta di un bound direzionale: la perturbazione di A dall'identità è piccola nella misura in cui agisce sulle direzioni dei moti osservati. Questo non implica che $\|A-I\|_2$ sia piccolo in senso spettrale, ossia uniformemente in tutte le direzioni di $\mathbb{R}^d$.

Il passaggio da controllo direzionale a controllo spettrale richiede ciò che nella letteratura va sotto il nome di condizione di copertura: le direzioni $\{\hat{v}_t\}$ esplorate durante l'addestramento devono coprire lo spazio latente in modo sufficientemente denso da che ogni direzione di $\mathbb{R}^d$ sia approssimabile con una combinazione delle direzioni visitate. Formalmente, se i versori $\{\hat{v}_t\}$ ε-coprono la sfera unitaria S^{d-1} — ossia per ogni $w \in S^{d-1}$ esiste t tale che $\|\hat{v}_t - w\|_2 \leq \varepsilon_{cov}$ — allora il bound direzionale si estende a un bound spettrale con una degradazione controllata dalla costante di copertura.

A7.4 La questione della copertura: un problema aperto di fondamentale importanza

Il Remark C.10 del paper di Wang et al. (2026) segnala con chiarezza, ma senza svilupparlo, il problema della transizione tra il bound direzionale e quello spettrale. Si riporta prima l'enunciato testuale del Remark, poi l'approfondimento.

Osservazione C.10 — Controllo direzionale vs. controllo spettrale (Wang et al., 2026). La Proposizione C.9 fornisce un bound su $(A - I)$ solo lungo le

direzioni visitate {v̂_t}. Estendere questo risultato a un bound spettrale uniforme ε = ‖A − I‖₂ richiede un'ulteriore condizione di copertura, affinché le direzioni visitate abbiano l'ampiezza sufficiente a coprire lo spazio latente. Questa non è un'ipotesi irrealistica in quanto le traiettorie di addestramento vengono tipicamente raccolte con sufficiente diversità. Sotto tali regimi, massimizzare la cosine similarity costituisce un proxy significativo per portare A vicino a I in norma spettrale.

Vale la pena di approfondire questa questione, perché rappresenta la principale lacuna teorica del framework attuale e individua con precisione dove si situa la frontiera del noto.

Il problema formale. Si supponga di aver verificato empiricamente che la cosine similarity media $\mathscr{C} \geq 1 - \eta$ su un corpus di traiettorie di addestramento. La Proposizione A7.3 garantisce che ‖(A−I)v̂_t‖₂ sia piccolo per i versori v̂_t osservati. Ma il bound sul condizionamento del Gramiano — l'equazione (A7.6) — richiede che ‖A−I‖₂ = ε sia piccolo in senso spettrale, il che implica che la perturbazione (A−I) sia piccola in qualsiasi direzione, non solo nelle direzioni visitate.

In termini geometrici, la situazione è la seguente. L'encoder apprende una traiettoria nell'intorno di un sottospazio effettivo dello spazio latente — quello generato dalle direzioni {v̂_t} — e il temporal straightening garantisce che la dinamica lineare approssimante sia quasi l'identità su questo sottospazio. Ma le direzioni ortogonali a questo sottospazio — quelle non esplorate durante l'addestramento — potrebbero corrispondere a valori singolari di A molto diversi dall'unità, senza che ciò si rifletta nella cosine similarity delle traiettorie osservate.

Una stima del gap. Si può quantificare il potenziale gap tra bound direzionale e bound spettrale nel modo seguente. Sia $\mathscr{V} := \text{span}\{\hat{v}_t : t = 0, ..., K\text{-}2\}$ il sottospazio generato dai versori di velocità, con dimensione $r \leq d$. Il bound direzionale controlla $\|(A-I)|_{\mathscr{V}}\|_F$, la norma di Frobenius della restrizione di (A−I) al sottospazio $\mathscr{V}$. Il bound spettrale ‖A−I‖₂ è la norma dell'operatore su tutto $\mathbb{R}^d$. Il rapporto tra questi due oggetti dipende dalla struttura di (A−I) sul complemento ortogonale $\mathscr{V}^\perp$: se (A−I) agisce come zero su $\mathscr{V}^\perp$, i due bound coincidono; se agisce come un operatore di grande norma su $\mathscr{V}^\perp$, il gap è arbitrariamente grande.

Questo argomento mostra che la condizione di copertura non è meramente tecnica ma è sostanziale: senza di essa, l'esistenza di direzioni latenti non esplorate durante l'addestramento può rendere il bound (A7.6) inapplicabile anche quando la cosine similarity osservata è alta. Per i sistemi di controllo robotico reali, in cui l'agente esplora una frazione limitata dello spazio degli stati, questa questione è genuinamente rilevante e merita attenzione nella progettazione dei protocolli di raccolta dati.

Una condizione sufficiente praticabile. Un'ipotesi che rende la transizione direzionale-spettrale formalmente controllabile è la seguente: si assume che la matrice di transizione A abbia struttura di basso rango più perturbazione isotropa, ossia $A = I + U \Sigma U^\top + \delta I$, dove $U \in \mathbb{R}^{d \times r}$ è una base ortonormale del sottospazio attivo e $\delta \geq 0$ è un termine di shift isotropo. In questo caso $\|A - I\|_2 = \max(\|\Sigma\|_2, \delta)$, e il controllo di $\|(A-I)\hat{v}_t\|_2$ per $\{\hat{v}_t\}$ che coprono $\mathcal{V} = \mathrm{im}(U)$ controlla $\|\Sigma\|_2$, mentre δ è identificabile separatamente dalla struttura isotropa del residuo. Questa decomposizione strutturale è consistente con l'osservazione che le dinamiche robotiche in ambienti semplici hanno tipicamente un numero ridotto di gradi di libertà attivi, rendendo l'ipotesi di basso rango plausibile in quei contesti.

La formalizzazione rigorosa di queste condizioni sufficienti e la derivazione del corrispondente bound spettrale costituiscono una direzione di ricerca aperta di notevole interesse teorico, che potrebbe produrre garanzie di condizionamento più forti e più facilmente verificabili empiricamente rispetto a quelle attualmente disponibili.

Appendice al Capitolo 7 — Evidenza sperimentale e raddrizzamento implicito

A7.5 Teleported PointMaze: un argomento empirico per la tesi informazionale

La tesi informazionale sviluppata nella sezione 7.9 afferma che il temporal straightening agisce come un collo di bottiglia selettivo che sopprime le informazioni ad alta variabilità angolare — irrilevanti per la prevedibilità della direzione delle transizioni — preservando quelle a coerenza direzionale temporale. Una difficoltà logica intrinseca a questa tesi è la seguente: negli

ambienti standard utilizzati negli esperimenti — Wall, PointMaze, PushT — la struttura visiva delle osservazioni e la struttura causale della dinamica sono correlate, perché stati visivamente simili tendono anche a essere geodeticamente vicini. In tali condizioni, è difficile distinguere empiricamente se un encoder che produce distanze euclidee allineate con le distanze geodetiche stia davvero catturando struttura causale o stia semplicemente sfruttando la prossimità visiva come proxy della prossimità dinamica.

L'esperimento del Teleported PointMaze, introdotto nell'Appendice E del paper di Wang et al. (2026), è specificamente progettato per rompere questa correlazione e permettere di discriminare tra le due ipotesi. La modifica introdotta rispetto al PointMaze standard è una dinamica di teletrasporto a senso unico: quando l'agente raggiunge il confine destro del labirinto — la regione evidenziata in rosso nella Figura 22 del paper — la sua posizione x viene istantaneamente reimpostata al confine sinistro, preservando la coordinata y e il valore assoluto della velocità lungo x. Formalmente, la regola di transizione modificata è

se $x_{t+1} > x_{\text{right-border}}$, allora $x_{t+1} \leftarrow x_{\text{left-border}}$,

$$y_{t+1} \leftarrow y_{t+1} \text{ (invariato)} ,$$

$$v_{x,t+1} \leftarrow |v_{x,t}| .$$

Questa modifica ha una conseguenza geometrica radicale: stati che si trovano nella stessa posizione visiva — prossimi al confine destro, indistinguibili dalle immagini RGB prodotte dall'ambiente — hanno distanze geodetiche drammaticamente diverse a seconda della posizione dell'obiettivo. Un agente vicino al confine destro che deve raggiungere un obiettivo a sinistra si trova in una situazione completamente diversa da quella di un agente nella stessa posizione che deve raggiungere un obiettivo a destra: nel primo caso, la strategia ottimale è attraversare il confine e sfruttare il teletrasporto; nel secondo, è semplicemente spostarsi verso destra senza attraversare il confine. La struttura visiva è identica nei due casi; la struttura causale è opposta.

La conseguenza per un encoder basato su prossimità visiva è immediata: esso assegnerà la stessa distanza euclidea a stati che dal punto di vista dinamico sono nella stessa condizione solo se il target è dalla parte sbagliata del teletrasporto — producendo un segnale di costo del tutto fuorviante per il

pianificatore. Le mappe di calore delle distanze nella Figura 23 del paper mostrano esplicitamente questa patologia per DINOv2: la mappa di distanza da un target fisso risulta quasi uniforme in tutta la regione visivamente omogenea vicino al confine destro, senza distinguere tra la zona pre-teletrasporto e quella post-teletrasporto.

L'encoder addestrato con temporal straightening, al contrario, produce una mappa di distanza che si allinea con la struttura causale: la regione vicina al confine destro presenta valori di distanza bassi (è geodeticamente vicina al confine sinistro grazie al teletrasporto) mentre la stessa regione, valutata rispetto a un target sul lato sinistro, mostra la corretta discontinuità causale. Questo risultato — visibile nella Figura 23(b) del paper a confronto con la 23(d) della ground truth calcolata con A-star — fornisce la prova sperimentale più diretta e convincente della tesi informazionale: l'encoder raddrizzato ha appreso a rappresentare la struttura causale delle transizioni, non la struttura visiva delle osservazioni.

Il confronto delle traiettorie di pianificazione nella Figura 24 del paper completa il quadro. Con temporal straightening, l'agente pianifica correttamente una traiettoria che attraversa il confine destro, sfrutta il teletrasporto, e raggiunge il target in un numero di passi consistente con il piano ottimale. Senza temporal straightening, l'agente si blocca nell'angolo vicino al confine, incapace di riconoscere che attraversare la parete è la mossa corretta — perché il suo spazio latente mappa la vicinanza del confine con la vicinanza visiva al target, che è nulla, anziché con la vicinanza causale, che è massima.

Implicazioni per la formalizzazione del collo di bottiglia. L'esperimento del Teleported PointMaze consente di precisare la formulazione informazionale della sezione 7.9 in modo più puntuale. L'encoder DINOv2 preserva un'alta mutua informazione $I(Z_t; O_t)$ tra l'embedding e l'osservazione visiva corrente. In un ambiente con struttura causale disallineata dalla struttura visiva, questa informazione è pianificazione-irrilevante o addirittura pianificazione-fuorviante: essa porta segnali di prossimità che inducono il pianificatore a ignorare le transizioni causalmente favorevoli. L'encoder raddrizzato, avendo ridotto questa componente visiva dell'informazione a favore della componente causale, opera secondo la corretta decomposizione del collo di bottiglia: $I(Z_t;$ struttura-causale$)$ è alta, mentre $I(Z_t;$ struttura-visiva-irrilevante$)$ è soppressa.

In termini quantitativi, il gap tra le mappe di distanza di DINOv2 e del modello raddrizzato rispetto alla ground truth A-star è misurabile come correlazione di Spearman tra i valori di distanza nelle due mappe e nella ground truth. Sebbene questa correlazione non sia riportata esplicitamente nel paper, la struttura visiva delle mappe della Figura 23 suggerisce che la correlazione dell'encoder raddrizzato con la ground truth sia sostanzialmente più alta di quella di DINOv2, in particolare nella regione prossima al confine di teletrasporto dove la discrepanza è più marcata.

A7.6 Il raddrizzamento implicito come pressione geometrica dell'obiettivo JEPA

Un fenomeno riportato nella sezione 5.2 del paper — e discusso nella monografia con accenni nella sezione 5 — merita un'analisi teorica approfondita per il carattere illuminante che assume rispetto all'intera architettura concettuale del temporal straightening. Il fenomeno è il seguente: anche senza l'aggiunta esplicita del termine di regolarizzazione di curvatura $\lambda \mathcal{L}_{curv}$, il solo addestramento del predittore con l'obiettivo JEPA produce una riduzione della curvatura delle traiettorie latenti rispetto al punto di partenza degli encoder DINOv2 non addestrati. In altre parole, il raddrizzamento non richiede necessariamente una regolarizzazione esplicita: esso emerge come effetto secondario, o effetto collaterale strutturale, dell'apprendimento predittivo.

Wang et al. (2026) attribuiscono questo effetto al fatto che l'obiettivo JEPA «favorisce rappresentazioni la cui evoluzione temporale sia facile da prevedere, così la pressione di addestramento riduce i cambiamenti direzionali bruschi nella traiettoria latente». Questa spiegazione informale è corretta nella sostanza, ma lascia aperta la questione di quale sia il meccanismo preciso attraverso cui la minimizzazione della perdita di previsione induce geometria retta. Vale la pena di sviluppare un argomento teorico più strutturato.

Un'analisi del gradiente per la rettitudine implicita. Consideriamo l'obiettivo di previsione nella sua forma standard senza stop-gradient per semplicità:

$$\mathcal{L}_{pred}(\phi, \theta) = \mathbb{E}\left[\| f_\theta(z_t, a_t) - z_{t+1} \|_2^2\right],$$

dove z_t = E_ϕ(o_t) e z_{t+1} = E_ϕ(o_{t+1}). Il gradiente di $\mathcal{L}$_{pred} rispetto ai parametri dell'encoder ϕ è

$$\nabla_\phi \mathcal{L}_{pred} = \mathbb{E}\,[2(f_\theta(z_t, a_t) - z_{t+1})^\top \cdot \partial(f_\theta(z_t, a_t) - z_{t+1})/\partial\phi] ,$$

dove il termine $\partial(f_\theta(z_t, a_t) - z_{t+1})/\partial\phi$ si decompone in due contributi: il gradiente attraverso z_t (che entra nel predittore come input) e il gradiente attraverso z_{t+1} (che entra come target). Concentrando l'attenzione sul contributo attraverso z_t, si ha

$$\partial f_\theta(z_t, a_t)/\partial\phi = [\partial f_\theta/\partial z_t] \cdot [\partial z_t/\partial\phi] = J_f(z_t) \cdot J_E(o_t) ,$$

dove J_f(z_t) := $\partial f_\theta/\partial z$ è lo Jacobiano del predittore rispetto allo stato latente e J_E(o_t) := $\partial E_\phi/\partial\phi$ è lo Jacobiano dell'encoder rispetto ai suoi parametri.

La direzione di aggiornamento del predittore f_θ tende a ridurre la norma del residuo $\|f_\theta(z_t, a_t) - z_{t+1}\|_2$. Questo significa che f_θ si adatta per mimare la mappa z_t → z_{t+1}: per ogni coppia (z_t, a_t) osservata, il predittore apprende a collocare f_θ(z_t, a_t) vicino a z_{t+1}. Nella misura in cui il predittore è una funzione regolare — un transformer o una MLP — esso ha un'inductive bias verso mappe con Jacobiano di norma controllata, ossia mappe che variano lentamente. Un predittore che impara a riprodurre fedelmente la mappa z_t → z_{t+1} per una famiglia di traiettorie addestrate sviluppa un certo grado di invarianza locale: le sue predizioni cambiano poco quando z_t varia di poco nella direzione delle traiettorie osservate.

L'argomento per la rettitudine implicita. Ora si può costruire l'argomento. Si supponga che il predittore f_θ si sia adattato a riprodurre la mappa (z_t, a_t) → z_{t+1} con alta fedeltà sulle traiettorie di addestramento. Se l'encoder producesse traiettorie altamente curve — con v_{t+1} molto diverso da v_t in direzione — allora lo stesso vettore z_t in stati temporalmente consecutivi sarebbe associato a predizioni z_{t+1} che variano rapidamente in direzione. Questo richiederebbe al predittore di implementare una funzione con Jacobiano rispetto a z_t di norma elevata nella direzione dei vettori di velocità successivi, aumentando la complessità funzionale della mappa appresa e la sua difficoltà di ottimizzazione.

Al contrario, se l'encoder producesse traiettorie rette, la funzione $(z_t, a_t) \rightarrow z_{t+1}$ avrebbe una struttura quasi traslazionale: $z_{t+1} \approx z_t + Ba_t$, ossia la predizione è quasi un'identità più un termine lineare nell'azione. Questa è la funzione più semplice possibile dal punto di vista della complessità funzionale, e il predittore convergente alla sua minimizzazione è naturalmente incentivato a operare in un regime dove l'encoder produce questa struttura semplice.

In termini più formali, il bias verso la rettitudine implicita è una conseguenza dell'inductive bias del predittore verso funzioni di bassa complessità. Se si parametrizza la complessità funzionale attraverso la norma spettrale del Jacobiano $\|J_f(z_t)\|_2$, i metodi di regolarizzazione implicita come la discesa stocastica del gradiente su reti neurali tendono a selezionare soluzioni con norma del Jacobiano ridotta tra tutte quelle compatibili con la perdita di addestramento. Una soluzione con norma del Jacobiano ridotta in tutte le direzioni delle traiettorie è precisamente quella in cui le transizioni sono quasi uniformi — quasi rettilinee — poiché un Jacobiano vicino all'identità produce, per input z_t che si muove lungo le traiettorie, output z_{t+1} che si muove nella stessa direzione con la stessa velocità.

Quantificazione dell'effetto. I dati sperimentali della Figura 5 del paper consentono di quantificare il grado di raddrizzamento implicito. Per gli ambienti Wall e PointMaze-UMaze, la cosine similarity media delle traiettorie prodotte dall'encoder con proiettore e $\lambda = 0$ (solo obiettivo di previsione, senza regolarizzazione di curvatura) è sistematicamente superiore alla cosine similarity del backbone DINOv2 non addestrato, di una quantità che varia approssimativamente tra 0.1 e 0.3 in unità di cosine similarity. Questo incremento non banale — prodotto dall'obiettivo JEPA da solo — costituisce il raddrizzamento implicito.

Il raddrizzamento esplicito — aggiungendo $\lambda > 0$ — produce un ulteriore incremento di cosine similarity di ampiezza comparabile, con un andamento che dipende dall'ambiente e dal tipo di encoder. Questo suggerisce che l'obiettivo JEPA da solo cattura circa la metà del beneficio geometrico raggiungibile, e che la regolarizzazione esplicita di curvatura completa la correzione verso lo spazio latente ottimale per la pianificazione.

Implicazioni per la progettazione. Il fenomeno del raddrizzamento implicito ha un'implicazione pratica rilevante: in regime di dati scarsi, dove la regolarizzazione esplicita di curvatura potrebbe introdurre rumore eccessivo nell'addestramento, è possibile ottenere parte del beneficio del temporal

straightening semplicemente addestando un predittore JEPA con sufficiente capacità e regolarizzazione implicita. Il contributo del termine esplicito è additivo e complementare: esso porta il modello più vicino all'ottimo geometrico, ma non è l'unico meccanismo attraverso cui la rettitudine emerge. Questa molteplicità di meccanismi di raddrizzamento — implicito ed esplicito — rende il temporal straightening un principio di regolarizzazione robusto, efficace anche in condizioni in cui la regolarizzazione esplicita è applicata con parametri λ subottimali.

A7.7 Raddrizzamento implicito nei modelli linguistici: una convergenza epistemica

Un punto di connessione intellettualmente rilevante, segnalato in modo cursorio nel paper e meritevole di un'analisi più distesa, è l'osservazione che il raddrizzamento implicito non è un fenomeno peculiare dei modelli del mondo visivi ma è stato documentato in architetture completamente diverse, in particolare nei modelli linguistici autoregressivi.

Il lavoro di Hosseini e Fedorenko (2023), citato da Wang et al. (2026), documenta che i modelli linguistici di grandi dimensioni ottimizzati per la previsione del token successivo producono, negli strati profondi, rappresentazioni delle frasi le cui traiettorie attraverso lo spazio delle attivazioni sono significativamente più rettilinee di quanto accade negli strati superficiali o in architetture non addestrate. Questo risultato, ottenuto su sequenze testuali anziché visive e attraverso un obiettivo di previsione linguistica anziché visuo-motoria, suggerisce che il raddrizzamento implicito sia una proprietà generale della previsione a sequenze: ogni sistema che impara a predire lo stato futuro da quello corrente tende, per la struttura dell'obiettivo, a sviluppare uno spazio di rappresentazione in cui l'evoluzione temporale è più rettilinea.

Il paper successivo di Hosseini et al. (2026) — «Context structure reshapes the representational geometry of language models» — approfondisce questa osservazione mostrando come la struttura contestuale del testo modifichi la geometria delle rappresentazioni: sequenze con struttura causale forte (frasi con dipendenze sintattiche a lungo raggio) producono traiettorie più rettilinee rispetto a sequenze casuali. Questo risultato è l'analogo linguistico di ciò che il Teleported PointMaze dimostra per le traiettorie visuo-motorie: la struttura causale del dominio si riflette nella geometria delle rappresentazioni apprese, e i

modelli addestrati sulla previsione della struttura causale sviluppano spazi latenti in cui questa struttura è rispecchiata metricamente.

La convergenza di questi risultati — modelli del mondo visivi, modelli linguistici, entrambi addestrati con obiettivi di previsione sequenziale — verso lo stesso fenomeno di raddrizzamento implicito suggerisce che tale fenomeno sia una proprietà universale dell'apprendimento predittivo su sequenze con struttura causale. Questa universalità dà al temporal straightening un significato che va oltre il dominio specifico della pianificazione robotica: esso è la versione esplicita e deliberatamente amplificata di un principio che i sistemi di previsione tendono ad apprendere comunque, e che il temporal straightening porta fino alla sua forma limite attraverso la regolarizzazione di curvatura.

A7.8 Una nota sul legame tra raddrizzamento temporale e misure di informazione geometrica

La formalizzazione del collo di bottiglia informazionale nella sezione 7.9 utilizza la mutua informazione come misura fondamentale. Tuttavia, esiste un approccio alternativo — forse più direttamente connesso alla geometria delle traiettorie — che impiega misure di informazione geometrica definite attraverso la curvatura della varietà appresa.

In geometria differenziale dell'informazione, la curvatura di una famiglia parametrica di distribuzioni P_θ è misurata attraverso il tensore di curvatura di Fisher, definito a partire dalla metrica di Fisher $g_{ij}(\theta) = \mathbb{E}_{P_\theta}[\partial_i \log p(x|\theta) \cdot \partial_j \log p(x|\theta)]$. La curvatura riemanniana associata a questa metrica misura il grado di non-linearità delle direzioni di variazione della distribuzione al variare dei parametri θ.

Nel contesto dell'encoder, se si parametrizza la distribuzione delle traiettorie latenti P_θ attraverso i parametri dell'encoder θ, la curvatura riemanniana dello spazio dei parametri rispetto alla metrica di Fisher misura esattamente quanto la struttura distribuzionale delle traiettorie varia non-linearmente con i parametri. Un encoder che produce traiettorie rettilinee corrisponde a una famiglia di distribuzioni che varia quasi linearmente con θ nella direzione delle traiettorie — il che corrisponde a bassa curvatura riemanniana in quelle direzioni.

Questa connessione tra curvatura delle traiettorie latenti e curvatura riemanniana dello spazio dei parametri non è sviluppata formalmente nel paper, ma fornisce un legame concettuale tra il temporal straightening e la geometria dell'informazione che potrebbe costituire la base per una formalizzazione più ricca della relazione tra rettitudine e compressione informazionale. In particolare, l'equivalenza tra curvatura riemanniana nulla e parametrizzazione lineare della famiglia di distribuzioni — che è esattamente la condizione di un sistema gaussiano con media che varia linearmente con i parametri — suggerisce che gli encoder raddrizzati si comportino, localmente lungo le traiettorie, come modelli gaussiani con drift lineare: una classe di modelli per cui le garanzie di apprendimento statistico sono particolarmente sviluppate e potrebbero offrire una traiettoria verso risultati di generalizzazione per il temporal straightening che vanno oltre le garanzie di ottimizzazione attualmente disponibili.

Bibliografia

Nota metodologica

La presente bibliografia raccoglie l'insieme completo delle opere citate nel corso della monografia e delle sue appendici, spaziando dall'originale letteratura sui modelli del mondo e sull'apprendimento autosupervisionato fino alle più recenti contribuzioni sul raddrizzamento temporale delle traiettorie latenti e sulla pianificazione basata su modelli. I riferimenti sono ordinati alfabeticamente per cognome del primo autore, con indicazione completa di autori, titolo, sede di pubblicazione, anno e — ove disponibile — identificativo arXiv. Le opere che presentano versioni preprint e versioni pubblicate in sede di conferenza o rivista sono citate nella loro forma più recente e autorevole.

Assel, H. V., Ibrahim, M., Biancalani, T., Regev, A. e Balestriero, R. (2025). «Joint embedding vs reconstruction: Provable benefits of latent space prediction for self supervised learning». *Advances in Neural Information Processing Systems* (NeurIPS 2025).

Assran, M., Duval, Q., Misra, I., Bojanowski, P., Vincent, P., Rabbat, M., LeCun, Y. e Ballas, N. (2023). «Self-supervised learning from images with a joint-embedding predictive architecture». In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (CVPR 2023), pp. 15619–15629.

Assran, M., Bardes, A., Fan, D., Garrido, Q., Howes, R., Komeili, M., Muckley, M., Rizvi, A., Roberts, C., Sinha, K., Zholus, A., Arnaud, S., Gejji, A., Martin, A., Hogan, F. R., Dugas, D., Bojanowski, P., Khalidov, V., Labatut, P., Massa, F., Szafraniec, M., Krishnakumar, K., Li, Y., Ma, X., Chandar, S., Meier, F., LeCun, Y., Rabbat, M. e Ballas, N. (2025). «V-JEPA 2: Self-supervised video models enable understanding, prediction and planning». Preprint, arXiv:2506.09985.

Ba, J. L., Kiros, J. R. e Hinton, G. E. (2016). «Layer normalization». Preprint, arXiv:1607.06450.

Bagad, P. N. e Zisserman, A. (2025). «Chirality in action: Time-aware video representation learning by latent straightening». *Advances in Neural Information Processing Systems* (NeurIPS 2025), The Thirty-ninth Annual Conference.

Bai, Y., Tran, D., Bar, A., LeCun, Y., Darrell, T. e Malik, J. (2025). «Whole-body conditioned egocentric video prediction». Preprint, arXiv:2506.21552.

Balestriero, R. e LeCun, Y. (2025). «LeJEPA: Provable and scalable self-supervised learning without the heuristics». Preprint, arXiv:2511.08544.

Bar, A., Zhou, G., Tran, D., Darrell, T. e LeCun, Y. (2025). «Navigation world models». In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (CVPR 2025).

Bardes, A., Ponce, J. e LeCun, Y. (2022). «VICReg: Variance-invariance-covariance regularization for self-supervised learning». In *Proceedings of the International Conference on Learning Representations* (ICLR 2022).

Bardes, A., Garrido, Q., Ponce, J., Chen, X., Rabbat, M., LeCun, Y., Assran, M. e Ballas, N. (2024). «Revisiting feature prediction for learning visual representations from video». Preprint, arXiv:2404.08471.

Bengio, S., Vinyals, O., Jaitly, N. e Shazeer, N. (2015). «Scheduled sampling for sequence prediction with recurrent neural networks». *Advances in Neural Information Processing Systems* (NeurIPS 2015), vol. 28.

Bucy, R. S. e Kalman, R. E. (1961). «New results in linear filtering and prediction theory». *Journal of Basic Engineering*, 83(1), pp. 95–108.

Camacho, E. F. e Bordons, C. (2004). *Model Predictive Control*. Seconda edizione. Springer, Londra.

Chen, C.-T. (1999). *Linear System Theory and Design*. The Oxford Series in Electrical and Computer Engineering. Oxford University Press, terza edizione.

Chen, T., Kornblith, S., Norouzi, M. e Hinton, G. (2020). «A simple framework for contrastive learning of visual representations». In *Proceedings of the 37th International Conference on Machine Learning* (ICML 2020), PMLR 119, pp. 1597–1607.

Chen, X. e He, K. (2021). «Exploring simple Siamese representation learning». In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (CVPR 2021), pp. 15750–15758.

Chi, C., Xu, Z., Feng, S., Cousineau, E., Du, Y., Burchfiel, B., Tedrake, R. e Song, S. (2025). «Diffusion policy: Visuomotor policy learning via action diffusion». *International Journal of Robotics Research* (IJRR), 44(1–2).

Cramér, H. e Wold, H. (1936). «Some theorems on distribution functions». *Journal of the London Mathematical Society*, s1-11(4), pp. 290–294.

Dave, I., Gupta, R., Rizve, M. N. e Shah, M. (2022). «TCLR: Temporal contrastive learning for video representation». *Computer Vision and Image Understanding*, 219, articolo 103406.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J. e Houlsby, N. (2021). «An image is worth 16×16 words: Transformers for image recognition at scale». In *Proceedings of the International Conference on Learning Representations* (ICLR 2021).

Du, Y., Yang, S., Dai, B., Dai, H., Nachum, O., Tenenbaum, J., Schuurmans, D. e Abbeel, P. (2023). «Learning universal policies via text-guided video generation». *Advances in Neural Information Processing Systems* (NeurIPS 2023), vol. 36.

Ebert, F., Finn, C., Dasari, S., Xie, A., Lee, A. e Levine, S. (2018). «Visual foresight: Model-based deep reinforcement learning for vision-based robotic control». Preprint, arXiv:1812.00568.

Epps, T. W. e Pulley, L. B. (1983). «A test for normality based on the empirical characteristic function». *Biometrika*, 70(3), pp. 723–726.

Eysenbach, B., Myers, V., Salakhutdinov, R. e Levine, S. (2024). «Inference via interpolation: Contrastive representations provably enable planning and inference». *Advances in Neural Information Processing Systems* (NeurIPS 2024).

Finn, C. e Levine, S. (2017). «Deep visual foresight for planning robot motion». In *Proceedings of the IEEE International Conference on Robotics and Automation* (ICRA 2017), pp. 2786–2793.

Fu, J., Kumar, A., Nachum, O., Tucker, G. e Levine, S. (2020). «D4RL: Datasets for deep data-driven reinforcement learning». Preprint, arXiv:2004.07219.

Goswami, R. G., Bar, A., Fan, D., Yang, T.-Y., Zhou, G., Krishnamurthy, P., Rabbat, M., Khorrami, F. e LeCun, Y. (2025). «World models can leverage human videos for dexterous manipulation». Preprint, arXiv:2512.13644.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. e Smola, A. (2012). «A kernel two-sample test». *Journal of Machine Learning Research*, 13, pp. 723–773.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo, Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R. e Valko, M. (2020). «Bootstrap your own latent: A new approach to self-supervised learning». *Advances in Neural Information Processing Systems* (NeurIPS 2020), vol. 33, pp. 21271–21284.

Ha, D. e Schmidhuber, J. (2018). «World models». Preprint, arXiv:1803.10122. Versione estesa pubblicata in *Neural Computation*, 32(7), 2020.

Hafner, D., Lillicrap, T., Fischer, I., Villegas, R., Ha, D., Lee, H. e Davidson, J. (2019). «Learning latent dynamics for planning from pixels». In *Proceedings of the 36th International Conference on Machine Learning* (ICML 2019), PMLR 97, pp. 2555–2565.

Hafner, D., Lillicrap, T., Ba, J. e Norouzi, M. (2020). «Dream to control: Learning behaviors by latent imagination». In *Proceedings of the International Conference on Learning Representations* (ICLR 2020).

Hafner, D., Lillicrap, T., Norouzi, M. e Ba, J. (2021). «Mastering Atari with discrete world models». In *Proceedings of the International Conference on Learning Representations* (ICLR 2021).

Hafner, D., Pasukonis, J., Ba, J. e Lillicrap, T. (2023). «Mastering diverse domains through world models». Preprint, arXiv:2301.04104.

Hansen, N., Wang, X. e Su, H. (2022). «Temporal difference learning for model predictive control». In *Proceedings of the 39th International Conference on Machine Learning* (ICML 2022), PMLR 162, pp. 8387–8406.

Hansen, N., Su, H. e Wang, X. (2024). «TD-MPC2: Scalable, robust world models for continuous control». In *Proceedings of the International Conference on Learning Representations* (ICLR 2024).

Harrington, A., DuTell, V., Tewari, A., Hamilton, M., Stent, S., Rosenholtz, R. e Freeman, W. T. (2023). «Exploring perceptual straightness in learned visual representations». In *Proceedings of the International Conference on Learning Representations* (ICLR 2023).

He, K., Zhang, X., Ren, S. e Sun, J. (2015). «Deep residual learning for image recognition». Preprint, arXiv:1512.03385. Versione pubblicata in *Proceedings of CVPR 2016*, pp. 770–778.

He, K., Fan, H., Wu, Y., Xie, S. e Girshick, R. (2020). «Momentum contrast for unsupervised visual representation learning». In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (CVPR 2020), pp. 9729–9738.

He, M., Balestriero, R. e LeCun, Y. (2024). «Towards principled architecture design for pre-trained world models». In *Workshop on Self-Supervised Learning: Theory and Practice*, NeurIPS 2024.

Hendrycks, D. e Gimpel, K. (2016). «Gaussian error linear units (GELUs)». Preprint, arXiv:1606.08415.

Hénaff, O. J., Goris, R. L. T. e Simoncelli, E. P. (2019). «Perceptual straightening of natural videos». *Nature Neuroscience*, 22(6), pp. 984–991.

Hosseini, E. A. e Fedorenko, E. (2023). «Large language models implicitly learn to straighten neural sentence trajectories to construct a predictive representation of natural language». Preprint, arXiv:2305.08076.

Hosseini, E. A., Li, Y., Bahri, Y., Campbell, D. e Lampinen, A. K. (2026). «Context structure reshapes the representational geometry of language models». Preprint, arXiv:2601.22364.

Internò, C., Geirhos, R., Olhofer, M., Liu, S., Hammer, B. e Klindt, D. (2025). «AI-generated video detection via perceptual straightening». In *Proceedings of the Thirty-ninth Annual Conference on Neural Information Processing Systems* (NeurIPS 2025).

Jaynes, E. T. (1957). «Information theory and statistical mechanics». *Physical Review*, 106(4), pp. 620–630.

Kailath, T. (1980). *Linear Systems*. Prentice-Hall, Englewood Cliffs.

Kuang, Y., Dagade, Y., Rudner, T. G. J., Balestriero, R. e LeCun, Y. (2026). «Rectified LpJEPA: Joint-embedding predictive architectures with sparse and maximum-entropy representations». Preprint, arXiv:2602.01456.

LeCun, Y. (2022). *A Path Towards Autonomous Machine Intelligence*. Documento di posizione, versione 0.9.2. Meta AI, giugno 2022. Disponibile al sito: openreview.net/forum?id=BZ5a1r-kVsf.

Levine, N., Chow, Y., Shu, R., Li, A., Ghavamzadeh, M. e Bui, H. (2020). «Prediction, consistency, curvature: Representation learning for locally-linear control». In *Proceedings of the International Conference on Learning Representations* (ICLR 2020).

Maes, L., Le Lidec, Q., Scieur, D., LeCun, Y. e Balestriero, R. (2026). «LeWorldModel: Stable End-to-End Joint-Embedding Predictive Architecture from Pixels». Preprint, arXiv:2603.19312v2 [cs.LG], 24 marzo 2026. URL: https://arxiv.org/abs/2603.19312.

Micheli, V., Alonso, E. e Fleuret, F. (2023). «Transformers are sample-efficient world models». In *Proceedings of the International Conference on Learning Representations* (ICLR 2023).

Nair, S., Rajeswaran, A., Kumar, V., Finn, C. e Gupta, A. (2022). «R3M: A universal visual representation for robot manipulation». In *Proceedings of the 6th Conference on Robot Learning* (CoRL 2022), PMLR 205.

Nguyen, D. e Widrow, B. (1989). «The truck backer-upper: An example of self-learning in neural networks». In *Proceedings of the 1989 International Joint Conference on Neural Networks* (IJCNN 1989), vol. 2, pp. 357–363.

Niu, X., Savin, C. e Simoncelli, E. P. (2024). «Learning predictable and robust neural representations by straightening image sequences». *Advances in Neural Information Processing Systems* (NeurIPS 2024), vol. 37.

Oh, J., Guo, X., Lee, H., Lewis, R. e Singh, S. (2015). «Action-conditional video prediction using deep networks in Atari games». *Advances in Neural Information Processing Systems* (NeurIPS 2015), vol. 28.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., Synnaeve, G., Xu, H., Jegou, H., Mairal, J., Labatut, P., Joulin, A. e Bojanowski, P. (2024). «DINOv2: Learning robust visual features without supervision». *Transactions on Machine Learning Research* (TMLR), 2024.

Peebles, W. e Xie, S. (2023). «Scalable diffusion models with Transformers». In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (ICCV 2023), pp. 4195–4205.

Pérez, J., Marinkovic, P. e Barceló, P. (2021). «On the Turing completeness of modern neural network architectures». Preprint, arXiv:1901.03429v3. Versione rivista di «On the Turing completeness of modern neural network architectures», *Proceedings of ICLR 2019*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G. e Sutskever, I. (2021). «Learning transferable visual models from natural language supervision». In *Proceedings of the 38th International Conference on Machine Learning* (ICML 2021), PMLR 139, pp. 8748–8763.

Robine, J., Höftmann, M., Uelwer, T. e Harmeling, S. (2023). «Transformer-based world models are happy with 100k interactions». In *Proceedings of the International Conference on Learning Representations* (ICLR 2023).

Rubinstein, R. Y. (1997). «Optimization of computer simulation models with rare events». *European Journal of Operational Research*, 99(1), pp. 89–112.

Rubinstein, R. Y. e Kroese, D. P. (2004). *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation, and Machine Learning*. Springer, New York.

Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., Levine, S. e Brain, G. (2018). «Time-contrastive networks: Self-supervised learning from video». In *Proceedings of the IEEE International Conference on Robotics and Automation* (ICRA 2018), pp. 1134–1141.

Siméoni, O., Vo, H. V., Seitzer, M., Baldassarre, F., Oquab, M., Jose, C., Khalidov, V., Szafraniec, M., Yi, S., Ramamonjisoa, M. et al. (2025). «DINOv3». Preprint, arXiv:2508.10104.

Sobal, V., Zhang, W., Cho, K., Balestriero, R., Rudner, T. G. J. e LeCun, Y. (2025). «Learning from reward-free offline data: A case for planning with latent dynamics models». *Advances in Neural Information Processing Systems* (NeurIPS 2025).

Sontag, E. D. (1998). *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. Texts in Applied Mathematics, vol. 6. Springer, seconda edizione.

Sutton, R. S. (1991). «Dyna, an integrated architecture for learning, planning, and reacting». *SIGART Bulletin*, 2(4), pp. 160–163.

Terver, B., Yang, T.-Y., Ponce, J., Bardes, A. e LeCun, Y. (2025). «What drives success in physical planning with joint-embedding predictive world models?». Preprint, arXiv:2512.24497.

van den Oord, A., Vinyals, O. e Kavukcuoglu, K. (2017). «Neural discrete representation learning». *Advances in Neural Information Processing Systems* (NeurIPS 2017), vol. 30.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. e Polosukhin, I. (2017). «Attention is all you need». *Advances in Neural Information Processing Systems* (NeurIPS 2017), vol. 30, pp. 5998–6008.

Wang, Y., Bounou, O., Zhou, G., Balestriero, R., Rudner, T. G. J., LeCun, Y. e Ren, M. (2026). «Temporal straightening for latent planning». Preprint, arXiv:2603.12231v1 [cs.LG], 12 marzo 2026.

Watter, M., Springenberg, J. T., Boedecker, J. e Riedmiller, M. (2015). «Embed to control: A locally linear latent dynamics model for control from raw images». *Advances in Neural Information Processing Systems* (NeurIPS 2015), vol. 28.

Williams, G., Aldrich, A. e Theodorou, E. (2015). «Model predictive path integral control using covariance variable importance sampling». Preprint, arXiv:1509.01149.

Yang, Y. e Ren, M. (2025). «Memory storyboard: Leveraging temporal segmentation for streaming self-supervised learning from egocentric videos». In *Proceedings of the Conference on Lifelong Learning Agents* (CoLLAs 2025).

Yun, C., Bhojanapalli, S., Rawat, A. S., Reddi, S. J. e Kumar, S. (2020). «Are Transformers universal approximators of sequence-to-sequence functions?». In *Proceedings of the International Conference on Learning Representations* (ICLR 2020).

Zhang, M., Vikram, S., Smith, L., Abbeel, P., Johnson, M. J. e Levine, S. (2019). «SOLAR: Deep structured representations for model-based reinforcement learning». In *Proceedings of the 36th International Conference on Machine Learning* (ICML 2019), PMLR 97, pp. 7444–7453.

Zhou, G., Pan, H., LeCun, Y. e Pinto, L. (2025). «DINO-WM: World models on pre-trained visual features enable zero-shot planning». In *Proceedings of the 42nd International Conference on Machine Learning* (ICML 2025).

Zhu, J., Evtimova, K., Chen, Y., Shwartz-Ziv, R. e LeCun, Y. (2024). «Variance-covariance regularization improves representation learning». Preprint, arXiv:2306.13292.